Second Class Postage Paid at Menlo Park, CA
Address Correction Requested

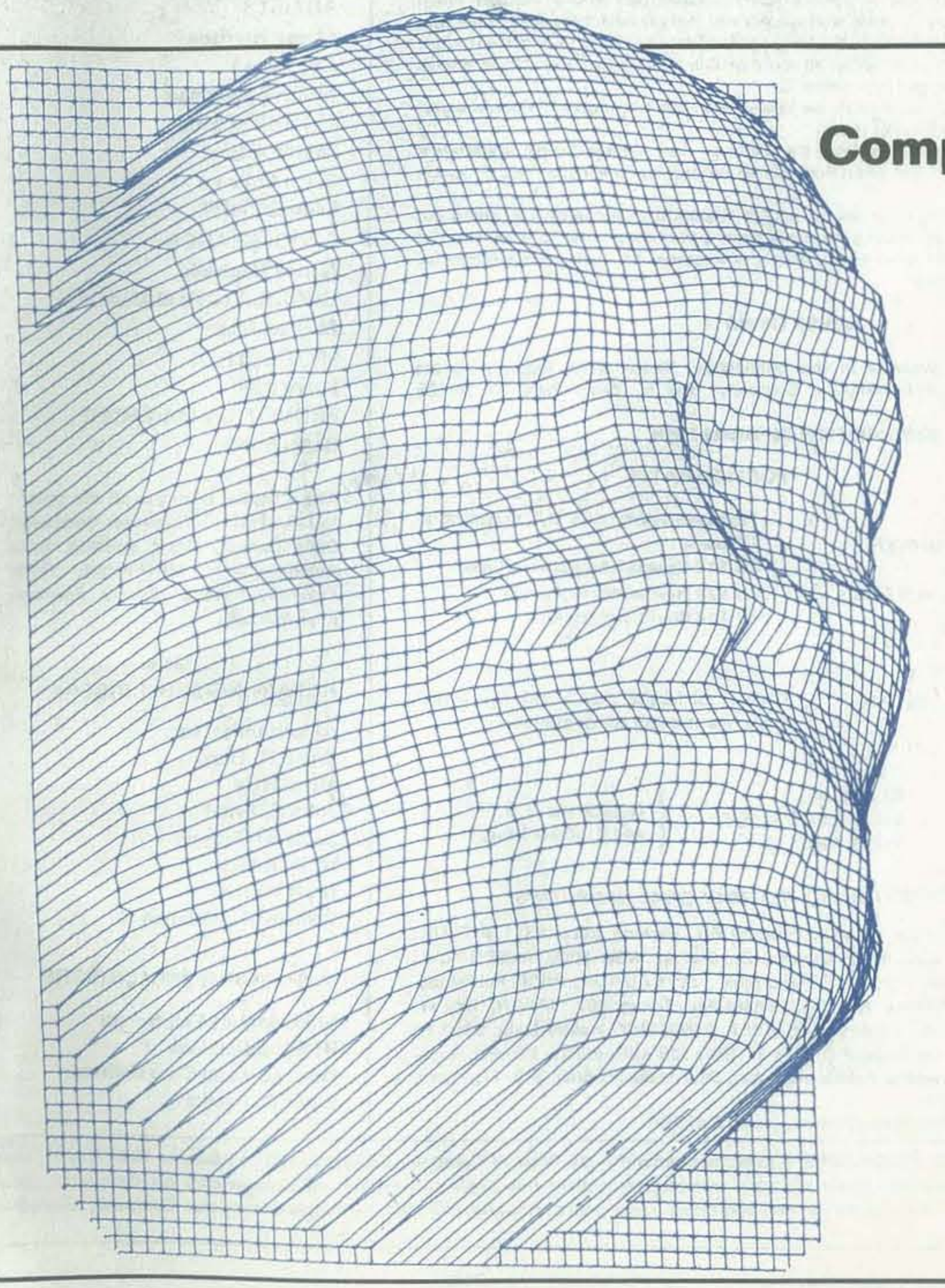
```
93065JOHNSB09          D1     2251
BYRON JOHNSON                 MY81
356 LAGUNA TERR
SIMI VALLEY, CA  93065
```

$2.00 In Canada $2.50

Recreational COMPUTING

FOR THE IMAGINATIVE SMALL COMPUTER USER!

VOL. 8 NO. 4 ISSUE 43 JANUARY-FEBRUARY 1980



Computing & Holistic Health

SUBMITTING ITEMS FOR PUBLICATION

LABEL everything with your name, address and the *date*; tapes should also include the program name, language and system. TYPE text if at all possible, double-spaced, on 8½x 11 inch white paper. DRAWINGS should be as clear and neat as possible in black ink on white paper.

LISTINGS are hard to reproduce clearly, so please note:
- Use a new ribbon on plain white paper when making a listing; we prefer roll paper or fan-fold paper.
- Send copies of one or more RUNs of your program, to verify that it runs and to provide a sense of how things work—and to motivate more of us to read the code. RUNs should illustrate the main purpose and operation of your program as clearly as possible. Bells, whistles and special features should just be described in the documentation unless they're particularly relevant.
- Make sure your code is well documented—use a separate sheet of paper. Refer to portions of code by line number or label or address, please, not by page number. When writing documentation, keep in mind that readers will include beginners and people who may be relatively inexperienced with the language you're using. Helpful documentation/ annotation can make your code useful to more people. Documentation should discuss just which cases are covered and which aren't.
- If you send us a program to publish, we reserve the right to annotate it (don't worry, we won't publish it if we don't like it).
- Last but not least, please try to limit the width of your listings: 50-60 characters is ideal. Narrow widths mean less reduction, better readability and better use of space.

LETTERS are always welcome; we assume it's OK to publish them unless you ask us not to. Upon request we will withhold your name from a published letter, but we will not publish correspondence sent to us anonymously. We reserve the right to edit letters for purposes of clarity and brevity.

ADVERTISING

ADVERTISING space is available in this publication. Please direct inquiries to the Advertising Manager, People's Computer Company, Box E, Menlo Park, CA 94025. (415) 323-3111

SUBSCRIPTION INFORMATION

U. S. RATES

☐ $10/1 yr. (6 issues)
☐ Retaining subscription @ $25 ($15 tax deductible)
☐ Sustaining subscription @ $100+ ($90 tax deductible)
☐ Lifetime subscription @ $900+ ($700 tax deductible)

Please allow 6-9 weeks for your first issue to arrive.

FOREIGN RATES

Payments must be in $US drawn on a US bank.
☐ $17 Canada/Mexico First Class
☐ $23 Rest of World Airmail
☐ $14 World Surface Mail

Delivery of foreign mail is slow and unreliable. We strongly advise airmail.

BACK ISSUES

$2.50 each:
Vol. 6, Nos. 1, 2, 3, 4, 5
Vol. 7, Nos. 1, 2

$3.00 each:
Vol. 7, Nos. 3, 4, 5, 6
Vol. 8, Nos. 1, 2, 3

Outside the U. S. add $ .50 per issue.

FOREIGN DISTRIBUTORS OF *RECREATIONAL COMPUTING*

LP Enterprises, 8/11 Cambridge House, Cambridge Rd., Barking, Essex 1G11 8NT, UK; **Electronic Concepts PTY Ltd.,** 52-58 Clarence St., Sydney, NSW 2000, AUSTRALIA; **RS-232,** 186 Queen St., W. Toronto, Ontario M5V 1Z1, CANADA; **ASCII Publishing,** 305 HI TORIO, 5-6-7 Minami Aoyama, Minato-Ku, Tokyo 107, JAPAN; **Ing. W. Hofacker,** D815 Holzkirchen, Lindenstr. 8, WEST GERMANY; **Hobby Data,** S200 12 Malmo, SWEDEN; **Computer Store,** P.O. Box 31-261, 22B Milford Rd., Milford, Auckland, NEW ZEALAND; **Systems Formulate Corp.,** Shin-Makicho Bldg. 1-8-17, Yaesu, Chuo-Ku, Tokyo 103, JAPAN.

Printed by Nowels, Menlo Park, CA

STAFF

EDITORS
Bob Albrecht
Tracy Deliman
Don Inman
Ramon Zamora
PUBLISHER
Willard J. Holden
ASSISTANT PUBLISHER
Sara Werry
PRODUCTION MANAGER
Carole Cullenbine
ARTISTS
Aleeca Harrison
Ann Miya
Judith Wasserman
TYPESETTERS
Phyllis Adams
Gavin Cullen
Laura Mendell
PROOFREADER
Nancy Heubach
CIRCULATION MANAGER
Michael Madaj
SPOT EDITOR
Harry Saal
PROMOTION MANAGER
Betsy Roeth

And a special thanks to all the other folks at People's Computer Company: Delia Daniels, Grant Groberg, Erica Sanders, Ann Merchberger, Nette Cekansky Wang, Sahnta Pannutti, Ellen Markoff.

RETAINING SUBSCRIBERS

Algorithmics, Inc.
David R. Dick
Mark Elgin
John B. Fried
Scott B. Guthery
W. A. Kelley
Brett Wilson
William M. Richman II

SUSTAINING SUBSCRIBERS

Bill Godbout Electronics
Byte Publications
Paul, Lori and Tom Calhoun
Louis R. Patzke

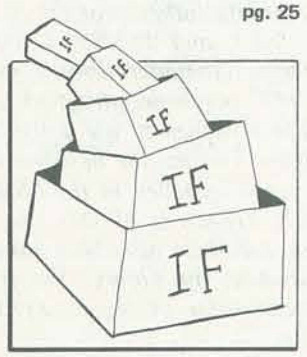
Recreational Computing (ISSN #0164-5846) is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a tax-exempt, independent, non-profit corporation, and donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points. Copyright © 1980 by People's Computer Company, Menlo Park, California.

Recreational COMPUTING

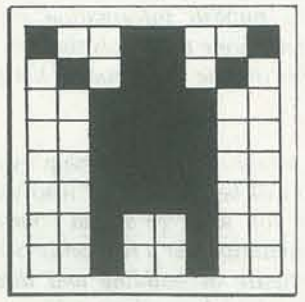
Volume 8 Number 4
January - February 1980
formerly people's computers

pg. 32

pg. 25



pg. 34


pg. 20

COVER by David Dameron

Editors' Notes

THE EDITORS' NEW YEAR NOTES

It's a new year! Time to make our resolutions, and wish ourselves a happy anniversary. This issue of Recreational Computing *marks the one year anniversary of the magazine's name change from* People's Computers. *There were other changes during the year. Louise Burton, one of the co-editors at the time of the name switch, has left the masthead, and two new co-editors have been added. Tracy Deliman joined the staff with the Sept./Oct. issue; Don Inman joined us in the Nov./Dec.* RC. *Bob Albrecht and Ramon Zamora, the other two co-editors, are still with us.*

*During the past year, several new features have been added to the magazine: Programmer's Toolbox*tm*, FuturePlay*tm*, Cryptarithms, The Outside Connection, Games For You To Program, and ongoing tutorials for the Atari and Texas Instruments computers. We've brought you over 100 articles, special features, and small programs during the past 12 months. This material came to us from all over the world, and was generally new copy not found in any other magazine.*

Beyond all the newness and changes, RC *remains committed to goals and principles that provided the motivation for starting the* People's Computer *magazine, and the* People's Computer Company Newspaper, *the two precursors of* Recreational Computing. *Those goals were most succinctly given by the Dragon, Bob Albrecht, way back at the start of our company:*

> Computers are mostly used against people instead
> of for people, used to control people instead of
> to free them. It's time to change all that—we need
> a People's Computer Company.

So PCC was founded as a non-profit educational corporation, with the aim of helping to put computers into the hands of people. Recreational Computing *is one publication of PCC, one vehicle to educate people about emerging computer technologies, and make that technology both useful and enjoyable for people. As the first periodical in the field of personal computing,* RC *is concerned with the future of computing and how it affects the individual.*

RC*'s readers are often its authors—its readers are its best and most frequent contributors.* RC *exists so our readers and the magazine can enjoy personal, social, and technological growth; we're also a forum for the interchange of ideas and issues.* RC *and its readers have been instrumental in fostering the growth of microcomputers as an industry and a personal tool.*

What about the future? The next year for RC *will be exciting. Each issue will have a specific theme. We'll bring you issues on the following topics: Games and Recreations; 6502 Processors; Simulations and Fantasy Role Playing; Learning; and Music and Art. We will be re-emphasizing games, recreation and challenges in all upcoming issues, since you've indicated that's what you want most. We will focus our coverage on five major micros—TRS-80, PET, APPLE, ATARI, and TI 99/4— with an increase in the number of short programs included. Our articles will be user-oriented material geared toward the beginner through intermediate level. The magazine's content will be broadened somewhat to allow expansion of our readership.*

RC *will have a new look. The cover will be upgraded to glossy paper, which will add four extra pages of content to the magazine and improve its appearance on the newstands.*

New columns and regular features will join the magazine in the coming year. With this issue, Bob Albrecht and Don Albers inaugurate a new challenge series, "Programming Problems," to run throughout the year. Dave Thornburg and Betty Burr are beginning a new column called "Computers and Society'" starting in the March/April issue of RC.

We will continue to report on the other projects of PCC, such as ComputerTown, U.S.A.!, and PCNET. ComputerTown U.S.A.! is an innovative computer literacy project, sponsored and peopled by PCC personnel, designed to give everyone in a town of 27,000 a chance to use a microcomputer. Bob Albrecht and Ramon Zamora, the initiators of the project now have four computers installed in the library of the town of Menlo Park, CA. Hundreds of kids have been validated to use the machines, and there have been numerous "open classroom" sessions held at the library. The project has enjoyed a great deal of local media coverage, and promises to gain some national attention.

PCNET, Personal Computer NETwork, is another project of PCC. PCNET offers personal computer-based electronic mail software. RC *plans to publish information on how PCNET works, some of the telephone protocols that are being used, and other aspects of this unique "grassroots" hardware/software project.*

RC *will also be conducting a survey of our readership by direct mail. The goal of the survey will be to find out who you are and what you do, and what you want to see in your magazine. The survey will be an invitation for a randomly-selected number of readers to participate in building and directing the future growth of the magazine. As we have said, our best material comes from our readers. We want to hear from all of you as we move into the future together.*

Since our main contributors are the readers of RC, we would like to invite you to send articles and programs. You can send us your material now for consideration for one of the theme areas in coming issues. Just note in your cover letter that your submission is to go into the "Learning" issue or the "Music and Art" issue. Of course, you can continue to send us material on any subject, as diversity is part of our character.

As a final note, we wish to thank the people who have supported us in the past year both by buying the magazine and by contributing material. We look forward to providing you for another year with a magazine that is one of the most content-packed periodicals on the market. As our name suggests, we will continue in the coming year to re-create the magazine with each issue, while looking both to the past and the future for inspiration. As for the computing part of our name, Bob Albrecht has noted, both to us and to you on several occasions, "Computing takes place in the mind, as well as in machines."

Ramon Zamora *Tracy Deliman*
Bob Albrecht *Don Inman*

Letters

NEEDS DENTAL SOFTWARE

I am the owner of a Radio Shack TRS-80. I presently have 32K, three disk drives, and the tractor feed printer. I am a general dentist, and would like to use the computer in my office. In case you are not familiar with the needs of a general dentist, I would like to do the following: 1) type patient name—screen shows name, address, etc., financial status, recall status, insurance information; 2) type employer—screen shows all details of insurance plan; 3) aged receivables—patient and insurance company; 4) appointment control and daily schedule; 5) fee list—A.D.A. code list; 6) insurance forms; 7) statements; 8) daily, weekly, monthly, and yearly reports; 9) provision for keeping track of production of multiple providers.

The system must be self-prompting, goof-proof, all data must be placed on the disk as it is entered. Disk and hard-copy backups must be automatic.

If you have this type of software, or are interested in creating it, please contact me immediately.

Stephen B. Katz, D.M.D.
1 Hoover Lane
New City, NY 10956
(914) 354-1181

THEY LOVE US!

I want to congratulate you on your November/December issue of *Recreational Computing*. We, the eighth graders of Whitehall Middle School and myself, have acquired three PETs by car washes, dances, suppers, and were just beginning to write music when Alfred Bruey's article appeared.

Keep the articles on the PET coming!

Also, I am interested in exchanging tapes with any junior high teachers who would like to exchange.

Gary W. Dode
Whitehall District Schools
Whitehall, MI 49461

COMPUTING IN EDUCATION

I am currently enrolled here at the University of Texas attempting to gain a degree in computer science. The computer seems the ideal means of eliminating much of the unnecessary portion of the educational system and allowing interested learners more direct access to the subject matter. Academia's attempts at self-preservation take no more grotesque forms than in the computer science department, I'm led to believe, after only two semesters back in school.

Yours seems like an organization with a lot of the same ideas. I reside on the ethnic side of town, and was appalled to learn that Chicano children are not given any instruction in Spanish. (Time was in Texas that children were forbidden to speak Spanish on school grounds.) My idea is to have lessons in Spanish, as well as subjects covered in school, programmed into a system. Individual students could get credit for completion, good for time on the games programs.

Keep up the good work.

Robert Garvey
Institute for the Advancement of
Absurdist Thought
1111 Willow Street
Austin, TX 78702

COMMENT ON PASCAL VS. BASIC

Just a note to say that I am amazed that no one has yet pointed out that Jim Day's BASIC fragment (Letters, May-June 1979) uses a *multiply*, and will execute much more slowly than any other PASCAL or BASIC fragment that has come under discussion in the "PASCAL vs. BASIC" articles.

Still working on Wylbur-Pilot, and you'll all be sorry when the technocratic empire crumbles.

Robert Solomon
181 Baltic Street
Brooklyn, NY 11201

HUMANIZING COMPUTER GAMES

So you play Star Trek and have logged a thousand hours as Captain of the *U.S.S. Enterprise*, but you need to keep a command sheet handy lest you forget # 9 is self-destruct. Why? You say it takes more memory and memory costs money. To use words to replace the sequence of number codes would use a little more memory but for command recognition it might be well worth it. I have never heard Captain Kirk order his action this way. Maybe it has not been done often because it would take away some of the fun when someone new sits down at our fancy machine to try his luck at the things we call fun.

Perhaps we would get more friends into the computer hobby if our games and programs were more *human*. It could be as simple as finding a new language for our machines or even to use the one we call BASIC to better advantage. As it is, we have programs that are far from understandable to outsiders. While this is fine with the simple Lunar Lander, it isn't good in a fantasy game or simulation.

I think we all can agree that a short program with a lot of remarks and advice is far more enjoyable for the newcomer and for ourselves. A program with a personality is well-liked and is played more than plain data-type programs. If you made two lunar lander programs, one with the bare bones and another with remarks like "You made a new crater _ KM deep," there would be little doubt which will get the most use.

In playing against the computer we often play against the perfect game or a game where the machine will make a mistake so you can win. This is why Star Trek is better—it is not played against the computer but instead is played against the clock and against an enemy that follows firm rules. It would be better still if the Klingons were controlled by another player rather than the computer, and also if they had movement within the entire galaxy instead of just one sector.

I am suggesting the interactive game where the computer is a moderator and scorekeeper only. If you have played Brett Tondreau's Galaxy II, you know what I mean. It is a space simulation where each player starts with one colony and from one to five ships, and through playing expands his holdings. The idea is to defeat all other players and rule the galaxy. You play one turn per month and get a total printout of your own actions and encounters. (If you want more data send 25¢ to Brett Tondreau, P.O. Box 7968, Van Nuys, CA 91409.)

Since Galaxy II is on a big machine it can only serve to give us ideas. Most home systems have but one terminal and most of our games are played alone or with one other player. We could use a modem to tie several systems together or even design a game where players are selected at random and we place our turn on cassette or disk and forward the media to the next player.

We could take a survey of a game like Galaxy II and use the data to design a single player game that could present a number of different playing styles, thus giving it a more human character. With a disk system many different playing styles and plots could be stored and accessed as needed. Even the sequence could be random.

Interaction between players in Galaxy II is what makes the game enjoyable. This is why the new crop of war games and fantasy games has grown to a reasonable portion of the hobby and craft industry in 1978. Any game played with other people is better than the perfect game played by the computer.

While I do not care for fantasy games as compared to science fiction games, I do think that games like Adventure are done well. These are some of the best games currently in use on microcomputers. Before I get all those fantasy freaks mad at me, let me explain.

I have read "The Lord of the Rings" and did enjoy it, but it is not real. To hide in this type of simulation is not realistic. Rather than waste my energy fighting fantasy with science fiction, I use my time on science fiction because I prefer to deal with things that could happen in future time.

In pondering this I realize some people will say that I am wrong too. OK, not all science fiction is possible but it is far more realistic to imagine oneself as a commander of a space ship, or ruler of a space empire that it is to place oneself in the form of some hobbit or wizard. You may then ask, "What about ESP?" The possibilities of ESP and the powers of the human mind are all accepted. As our horizons expand I am sure we will find stranger things yet. Part of the fun in Fantasy games is that they do not follow the normal rules of time, space and the universe.

While half of you now hate me and the other half says right on, I feel another group looking and saying games are a waste of time and serve no useful purpose. Remember, all work and no play is the reason so many people crack. Enough people do play games to make this subject worth talking about.

What now? Well, first we must find a good language for our games, or to design a new one to handle the special things we want to do in our simulations. We need some priorities: 1) we need a good conversational language like PILOT; 2) we need some math to cover calculations and scoring; 3) if graphics are to be used then we need a good fast way of implementing them.

In the actual game itself we need many things to make it mean something. It must: 1) have personality; 2) be one that we can win; 3) have a variety of characters; 4) have more than one playing strategy; 5) be easy for beginners to play.

If we design a game where more than one human player is involved then the computer should act as the moderator and the scorekeeper. It should aid any and all players in the same way.

Sometimes we need to stop and see where we have been and where we are going. We may find that we have missed the point and if we stop and look, we may find our way to the right track. Star Trek has served us well but it should be replaced by a new more human and civilized game. Of all of science fiction, Star Trek is the most humane, yet our game is barbaric with murder and death. We should strive to end this type of game and use our time to make games that help us to know other people better, and to understand them.

You may say that it's only a game. Yes it is, but war and politics can be considered games, too. Such games could very well put an end to the human race. We should all try to bring in a new group of games designed to make a better world for humanity, not make it easier to destroy man.

Live long and prosper.

Robert Howarth, Jr.
RFD # 1 Box 36
Lisbon, NH 03585

WHO CAN ANSWER?

Please send me information about PILOT. How can it be implemented on a PET? On a PDP-8? On a PDP-11? How costly is it? How much time does it take to receive it? How much training do teachers need to use it with students? For what reasons was it developed? Is there research supporting its usefulness? For what is it most useful? Are there other materials supporting its use in classrooms? Is PILOT in any way related to LOGO?

Additionally, I would like to know about your Society for PET Owners and Trainers.

Thank you!
Beth Lowd
Specialist, Computers in Instruction
Lexington Public Schools
9 Philip Road
Lexington, MA 02173

The Society for PET Owners and Trainers is a truly unique society, comprised of everyone who considers themselves a PET user. There are no meetings, no membership dues, no elections . . . in fact, no organization. Actually, it is the SPOT column by Harry Saal.

Cryptarithm Solutions (1979)

		Number/Letter Correspondence									
RC Issue	Puzzle No.	0	1	2	3	4	5	6	7	8	9
No. 40, July-Aug.	1	–	C	–	–	–	B	–	–	–	–
	2	–	T	–	–	–	–	–	–	–	R
	3	B	Y	C	A	H	–	F	–	–	J
	4	M	H	K	C	B	F	–	–	A	–
	5	D	M	–	–	–	–	–	–	H	K
	6	D	E	H	N	T	U	S	Y	A	B
	7	B	U	Y	E	S	M	L	N	A	R
	8	E	H	B	T	D	L	W	U	M	C
No. 41, Sept-Oct	9	–	M	–	–	–	–	–	–	L	S
	10	–	C	D	F	K	A	M	R	–	B
	11	Y	D	T	A	B	U	C	R	N	E
	12	A	B	E	H	L	M	S	T	U	Y

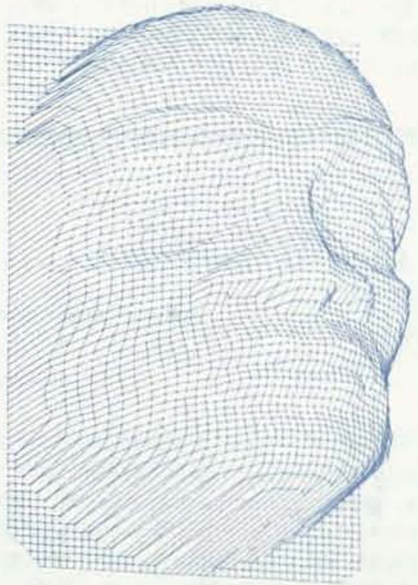
There are no Cryptarithms in this issue of RC. Look for Cryptarithms again in the March - April Games & Recreations issue of the magazine.

COMPUTING FOR HEALTH AND EQUALITY

BY DAVID J. HALL,
M.Sc. (Eng.) D.I.C.

David Hall is a senior research engineer who not long ago turned holistically healthy. Here, he presents a refreshing and unique approach to both health and computing. David parallels many of the ideals for health that come from the holistic health movement with some of the principles involved in computing. He views computing as a mind function as well as machine function, and shows the relevance of these ideas in computing for health. On the practical side, David describes his own computer programs on health and equality. Hopefully, David's explorations in this area may generate innovative programming ideas among our readers. The author's address is P.O. Box 9487, Stanford, CA 94305.
- TD

A good way to begin is with definitions. Computing means literally "thinking-with," that is, thinking or figuring out with a tool such as pencil and paper. Health is the condition of the human body, mind and spirit that enables it to perform its vital functions well. Thus "computing for health" is using our most modern tool, the computer, to maintain and improve our performance of vital functions. "Equality" is a term used in mathematical equations, arithmetic and computation. Equality is also used as a social ideal and it implies balance, an important property of health. The practical application of equality in social systems is currently important in the ideals of equal opportunities for men and women, and for all races. The healthily-used computer should maximize the whole function of living.



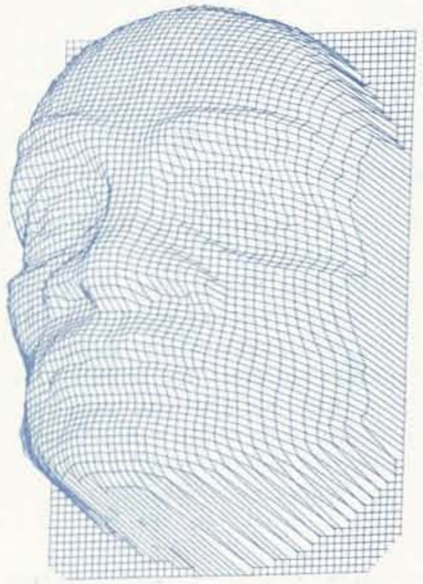
We readers of *Recreational Computing* do not need to be reminded how economical the price of computing has become over the last several years, so that vast computing power is now available at the domestic level. Perhaps health is our most precious asset, so it is natural for a computer engineer like me to ask: what can my computer or my thinking processes do to promote health? Can health be logically deduced and calculated, or is it a matter of heredity, good fortune and luck? As a senior research engineer with 20 years experience in electronics and computing and as a professional health worker for the last 6 years, I have given these topics much thought. Just as there is a new trend towards low-cost microprocessor technology in *computing*, there is also a new trend in *health*.

Holistic health is a recent popular trend that seems unorthodox compared to the approach of the established M.D. The basic idea of holism (a concept formulated by J.C. Smuts, the former prime minister of South Africa) is that the individual is treated as a whole entity, equal attention is given to all parts of the human being, including body, mind and spirit. Orthodox medical treatment, on the other hand, concerns itself with gross pathological problems that doctors seldom recognize unless they have measurable material effects. Orthodox treatment is administered by highly-specialized doctors practicing in non-overlapping fields, such as heart surgery or ear-nose-and-throat specialities.

Modern orthodox medicine has accepted the use of computers very well, perhaps too well. It seems that M.D.s have been discouraged from using their own hands because of the malpractice insurance problems implied in touching a patient. There seems to be an over-reliance on the use of sophisticated tools in modern orthodox medicine; it is apparently considered more scientific to test and measure than to feel. These complex tools and instruments provide only second-hand knowledge though, not the direct experience of "putting one's finger on the problem" that many of the holistic approaches encourage. While the wonderful, detailed pictures being produced by the latest computerized tomographic scanners are surely very useful to the brain or heart surgeon, and are a triumph of sophisticated computerized technology, most common health problems can be solved by a more simple and direct approach. This approach uses the trained human hand, as in rolfing, for example, a system of deep myofascial manipulation, or massage.

These are the two extremes of health care: first the remote physician does not physically contact his patient, but rather subjects him to many tests by instruments and to swallowing much medication (with numerous side effects) of recent invention by sophisticated chemists. Second, there is the more intimate healer, who personally observes and methodically contacts all of his client's important parts, and calls upon the client to heal himself by releasing undue stresses.

The small computer, being accessible domestically, is a good "middle-ground" tool. It does not replace the skilled human hand, or the highly technologized physician, but the computer is able to process effectively some of the data concerning the simpler health factors. Many simple peripheral monitoring devices for measuring bodily performance might eventually be devised for connection to a home computer. For example, perhaps a low-cost heartbeat monitor for displaying blood pressure may soon be available for connecting yourself to your home computer. In this article, we will examine how we can input data via the keyboard of a small computer (in this case, the PET), so that valuable information about maintaining or improving health can be computed and displayed.



WHAT HAS COMPUTING TO DO WITH HEALTH?

There is no doubt that computing has more impact on our lives today than ten years ago. We have witnessed the advent of many kinds of calculators, digital watches and personal computers. The purpose of technology should be to contribute to our comfort and well-being. With all the material resources available today, we have little excuse for being unhealthy, except that confusion tends to be rampant in a surfeit of resources. Computing of the right kind—counting what really matters, evaluating the data counted and verifying the programs that process that data—are activities that can help put order in our lives, contribute to clarity and thus lead to health. More specifically, we can use a computer to evaluate the details of basic factors related to health, such as diet, exercise, cleanliness and rest.

Low-cost computing can take many complex health factors into account, and can be accessible to a large number of users. After reliable software becomes available, family members should be encouraged to use a home computer for health checkups. It is likely that we will soon find many software developments in the area of domestic health, so that computing will eventually be of significant benefit to social health.

CHOICE OF CODING CONTROLS THE MACHINE ACCORDING TO HEALTHY DESIRES

In running any form of computer, an important issue is how the *coding* of one's wishes for certain characteristics of *input, processing*, and *output* can be easily accomplished. Operation codes (OPCODES) for each type of computer define "machine-language" properties. All other high-level languages invented for ease of human programming, must be coded-down to machine language before execution. A "healthy" machine is one which realizes the desired input, processing, and output functions, just as a healthy person is one who realizes his own desires. Then, just as a healthy attitude is supported by good posture, a well-constructed and reliable computer program needs good structure as in "structured programming."

COMPUTING THE SOLUTION TO A HEALTH PROBLEM

Let us assume that you have a small computer that is suitable for figuring out solutions (S) to health problems, using data (D), and suitable programs (P). Mathematically, we say that the solution is derived by evaluation of the function P(D). Or, the solution is equal to the program running with data. (See Figure 1 for a block-diagram or flowchart of this process):

$$S = P(D)$$

Although this equation is conceptually very simple, it has introduced notions and notations that are the basic symbols used in computation. Because this is an equation, to fully understand its meaning, we must understand the notion of *equality*: both in its strict *numerical* sense, and also in its *constitutional* sense. As a conscientious practitioner of the holistic discipline of rolfing, I interpret balance and equality of shapes. In rolfing this concerns placements of masses of human tissue, while in computing, it concerns the computational limits of numerical equality. Also, we must consider its opposite, inequality, which would be termed discrimination or classification in social or mathematical terms. The program "Equality" that I have developed for the PET covers both these social and computational aspects of equality. (See later description.) In the sections following, several *computing* ideas that relate to health or equality are discussed; we see what happens in testing for numerical equality on the PET. Socio-sexual roles in everyday living functions are examined for you or any family member. Other ideas about the arrangement of material tissue and socio-sexual roles are present. (This touches on the dress-coding of our bodies, since good clothing is an arrangement of materials about the body, fitting well, and serving the function the human body is performing.)

THE MEANING OF EQUALITY, NUMERICALLY

In the numerical sense, a reliable computer should always give the correct answer in testing for equality. Naturally, there are limits to the accuracy of any computer. These limits are demonstrated for the PET using the "Equality" cassette tape. On this tape, I show that PET's BASIC makes an error in testing for numerical equality, but this is not usually a serious error. The program tests the "mental" health of the PET system itself, showing that it should not always be believed, and the mental health of the user concerning arithmetic precision related to notions of equality. For example, when we RUN the following statement:

```
IF 99.99999999 = 99.999999999 THEN
PRINT "TRUE"
```

the PET prints TRUE, even though the second number has one more nine after the decimal point, which makes the second number larger. The problem arises because the machine can only deal with 10 digits of precision and it "forgets" digits beyond that. It should signal some kind of syntax error, but does not do that.

THE MATHEMATICAL SIGNIFICANCE OF EQUALITY APPLIED TO SOCIO-SEXUAL CUSTOMS

In the constitutional sense, equality is connected with freedom of opportunity and equal rights. Because we are social beings, our constitutional health applies both to our individual and collective bodies. Whatever social classification we are born into, we do not expect ideally to be penalized or discriminated against because of it. Any individual automatically belongs to both a racial and sexual classification, but these specifics should neither penalize nor restrict liberated behavior.

Do you have the freedom to act beyond your classification and stereotype boundaries? For example, if you are a man, would you feel comfortable washing dishes, or doing other traditionally womanly activities? Or, if you are from a black family, can you choose your home in the *white* suburbs? Equality tests some of these ideas as they apply to your daily life activities, and makes a light-hearted judgment as to your sex, based upon your answers.

Society has placed upon men and women a need for clothing. Our natural state is without clothing, therefore clothes should be considered from the aspect of health as well as from the aspect of fashion. For example, in Victorian times women often fainted because of tight corsets. Similarly some men today, restrict themselves with tight collars and ties. Clothing may differ according to sex differences but if society is balanced and equal, it wil not discriminate against the free choice of clothing for any *body*, because of its sex. If society is healthy and balanced, it will also be seen to be balanced by its appearance and this implies that a census or count of the frequency of the two basic garments (skirt or pants) on either sex, will be observed as approximately equal. Since function and structure are related in a complete, holistic consideration, the functions that men and women perform can be evaluated with respect to equality by observing what function each individual performs, for example, washing dishes, preparing food, repairing motor vehicles, gardening, etc.

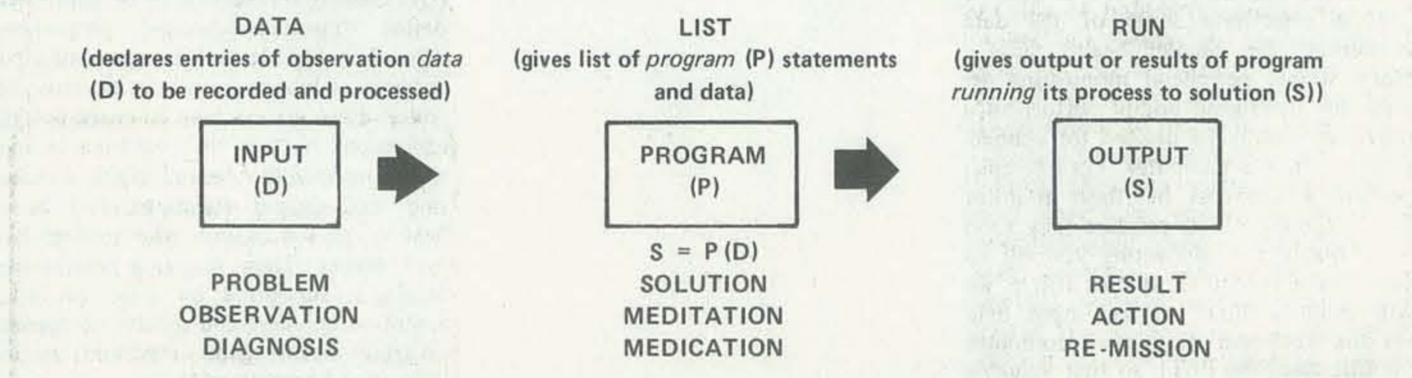

DATA
(declares entries of observation *data* (D) to be recorded and processed)
LIST
(gives list of *program* (P) statements and data)
RUN
(gives output or results of program *running* its process to solution (S))
INPUT (D)
PROGRAM (P)
OUTPUT (S)
S = P(D)
PROBLEM
OBSERVATION
DIAGNOSIS
SOLUTION
MEDITATION
MEDICATION
RESULT
ACTION
RE-MISSION

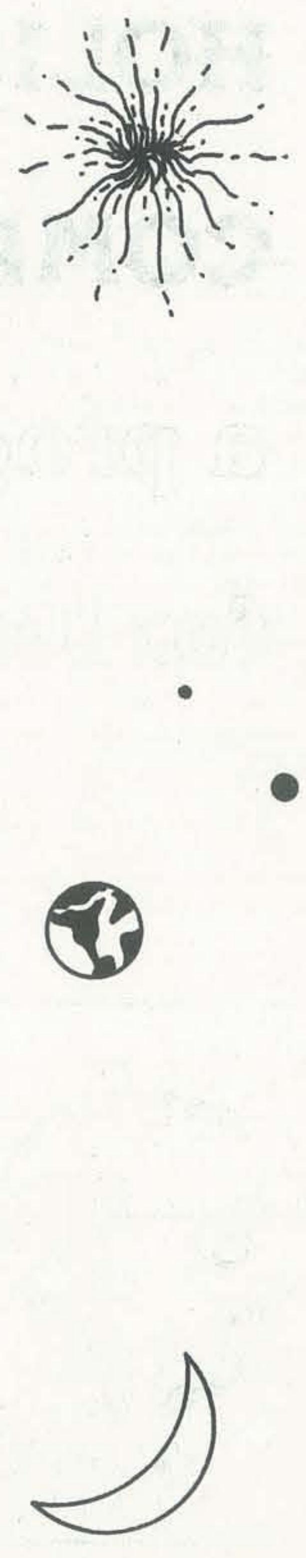
Figure 1 The Three Elements of Computing in Several Disciplines

The coding of machine instructions is like the soft-ware of the human body, that is, computer codes and dress codes affect the behavior and functioning of intelligence, whether artificial or real. Dress coding is usually decided by the fashion designers and clothing manufacturers, or by the public service organizations for which formal and recognizable uniforms are appropriate. (See my paper, listed in References).

PROGRAM PROVING

The important goal in using computers for health is to be sure that the program's results really do contribute to health rather than cause damage through error. Any healer, psychiatrist, M.D., yoga teacher, rolfer or other health practitioner must not cause harm, but should contribute to health. Therefore, results of treatment must be constantly checked and verified, as our ultimate measure of results is success. The deepest kind of success, however, is felt internally. We are certain that our computing machine does not feel its own degree of certainty, it merely acts without conscience. Evaluation of the success of a computer program for health then, is to be trusted to the human judgement of the user.

Health and intelligent behavior is a vast and mysterious topic. We can manipulate factual information about our health, and can give it our attention, and thus we may improve it, since *attention relieves tension*. There may always be a somewhat unsatisfactory or incomplete feeling about running a specific health-related program on a small computer when the user holds the whole concept and detail of health in mind. It becomes the responsibility of the health program user to constantly question the health program's validity, just as the computer questions your validity in checking your health when it asks you specific questions. What I have devised is a series of health-related computing exercises for the mind, via the keyboard and display. I hope these contribute to your health, but bear in mind that they cannot supply the magical ingredient that comes from right living in tune with natural order.



"Perhaps a low-cost heartbeat monitor for displaying blood pressure may soon be available for connecting yourself to your home computer."

SUMMARY OF HEALTH-RELATED PROGRAMS FOR THE PET

The subject of health is so vast that covering even a small part of it may require a library of applicable programs. The first three I have been developing are:

Health. This is an instructive program that presents some fundamental statements about health and asks some questions about your status and health factors. It is based on the "normal distribution" curve of de Moivre.

Unfortunately, we have not yet been able to fully harness our newly acquired and immense computing power for health programs in the home. The basic elements of health are cleanliness, diet, exercise, and rest. Each of these elements should be examined by each individual in the family once-per-week initially, but if illness occurs, then a daily check-up is better. A checkup begins with your current status.

The questions asked by the program are the simple physical variables such as age, weight, sex and height. Your particular measurements can be compared to a healthy average. This data comprises your current status or stature. Questions are asked about your lifestyle in relation to these four basic elements and a score is computed for each of these elements. The total score is a measure of your health.

Health introduces itself as shown below:

```
HEALTH
AN INTERVIEW BY
DR. PET TO
DIAGNOSE YOUR
HEALTH
HEALTH IS THE COMPLETE
COMBINATION OF
OPPOSITES, AS YIN
AND YANG.
```

Equality. This program deals with the notions of equality and discrimination. In BASIC terms, = is the symbol used to signify equality, whereas <> signifies its opposite, inequality. Tests of the limits of discrimination of equality in the PET are given, and its shown that some of the PET's conclusions are faulty, indicating that it is failing in certain areas of "mental health."

Note in the test for equality, that MAN < WOMAN, but that MALE > FEMALE. Similarly, BLACK < WHITE and NATURAL > CULTURED. Although these terms are ones of social equality, the computer does not know the meaning of them, only their alphanumeric precedence based on the ASCII code. However, it is useful that the machine can order any word-pair you might like to type as a test for equality in this way.

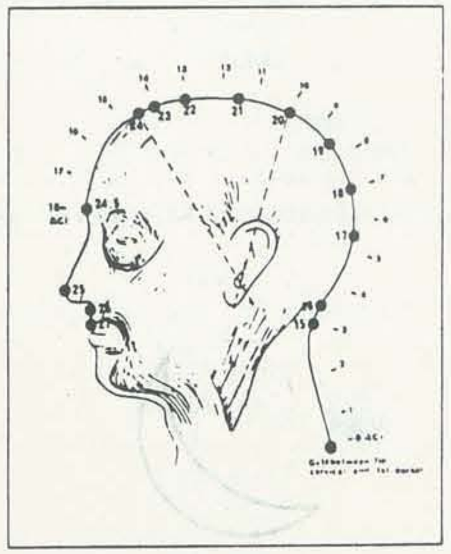
Telindex. This program is being developed for health referrals. It will list specific health professionals and organizations under their general headings, for example: M.D., podiatrist, rolfer, acupuncturist, dentist, etc. (These professionals will have paid a small fee to be listed on this tape, and they will be personally recommended as being excellent practitioners in their fields.) *Telindex* also has a more general function as a *tel*ephone *index* program that stores your frequently-used names and telephone numbers so that the number of the person you want to call can be instantly displayed on the screen. The input required to find a number is minimal, since usually the first few characters of the person's last name will uniquely identify that person. The program is intended to be run continuously while the PET is not needed for anything else. A *User's Guide for the Telindex Program* supplied with the cassette tape, will give further details of the program.

REFERENCES

1. Wischnia, Bob. "Rolfing" in *Runners' World*, January 1979.
2. Smuts, J.C. *Holism and Evolution*. New York: The Viking Press, Compass Books Edition, 1961.
3. Hall, David J. "Advantages and Disadvantages of Various Forms of Body Covering" (available from the author).

HOLISTIC COMPUTING

a program idea for healthy living



BY TRACY DELIMAN

This article describes the basics for a user-oriented computer program on holistic health, a program to help anyone maximize their efficiency and health throughout daily activities. The idea was inspired by my own studies in holistic health and by David Hall's article in this issue of RC. — TD

One of the visions that emerges from the humanistic growth and holistic health movements is the image of a kind of "super being." The vision varies somewhat, but basically this human has nearly boundless energy, is highly productive and creative, can handle responsibility and social demands well, and is peaceful and content in personality. Sounds ideal? Recently, many people are jogging more, cutting down on coffee, sugar and fats, meditating, and becoming more caring in their personal relationships. They're also becoming more productive and more content with their lives.

As I watch these events, I realize that the vision is actually not impossible to achieve, but that getting there requires a gradual increase of awareness of *all* the details of one's health.

Developing that awareness means observing, on a daily basis, several kinds of physiological responses which are known indicators of real bodily changes. These changes are associated with physical and mental activities, diet, mood or psychoemotional state, illness and fatigue, and attentiveness. If we thoroughly understood these responses in ourselves and could monitor them every day, we would be able to plan our work-productive and creative activities with much greater efficiency. Imagine how much more you could achieve and create if on each day you knew concretely which of a multitude of activities you were best suited to undertake on that day.

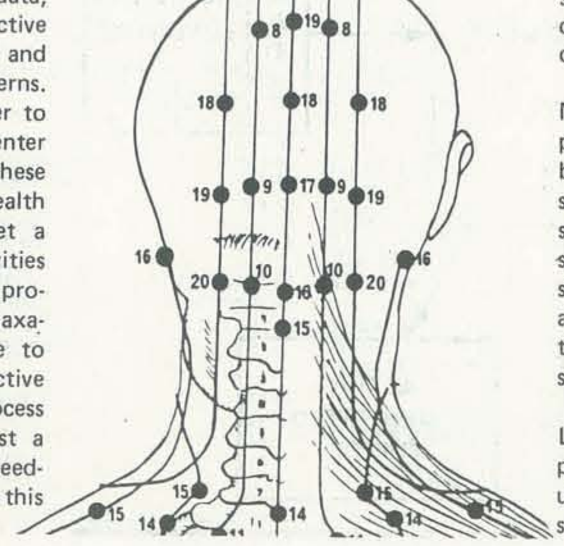
To a certain degree we already observe ourselves and sometimes make changes in dietary or other patterns so that we will function better or feel better. But to make such detailed observations everyday as would be necessary to really maximize our productivity and creativity would require years of training and careful self-observation. It seems obvious then that a reliable electronic device would be more accurate in objectively assessing, recording and correlating those subtle physiological responses.

Out of these musings and desires a program idea has emerged. A simple biofeedback machine would be interfaced to the computer (with an analog to digital converter to change the data to numbers that the computer can use). The program itself would include a personal profile as data base, detailing standard health data, a list of the individual's work-productive and creative activities, recreational and relaxation activities, and dietary patterns. This program would enable the user to connect with the machines and enter daily health data directly, have these scores correlated with standard health scores (for that person), then get a reading of which group of activities should be followed that day. The program would also display suggested relaxation activities and dietary schedule to complement the selected productive activities. This holistic computing process will be further elaborated, but first a closer look at what's involved in biofeedback that would be applicable to this program idea.

BIOFEEDBACK

The science of biofeedback already accomplishes much of the task of sensing and recording physiological data. Biofeedback is most commonly used for the purpose of learning to voluntarily control certain physiological and mental processes that were previously believed to be entirely involuntary. When an individual is connected with the biofeedback machine, physiological processes and responses are measured and fed back to the person. Then, with many hours of observation of these responses and with the guidance of a trained practitioner, an individual learns some voluntary control over these processes. The essential idea in biofeedback is to sensitize our self-observation abilities and learn self-control.

The biological processes usually measured in biofeedback training are skin temperature, galvanic skin response (GSR), brainwaves, muscle tension and eye movement. These responses relate to emotional changes, nervousness and attentiveness. Each of these processes is affected by changes in the environment, or by changes in the physiological or consciousness level. Stress or an altered mood, for example, can produce a response in one of these areas.



Galvanic skin response is the natural electrical resistance of the skin. Changes in this response are very subtle,and are an indication of arousal of the autonomic nervous system or of stress. A change in skin temperature is also an indicator of stress or discomfort, as most of us know from having experienced cold hands at a job interview or similar situation.

Brainwaves are more complex than most of these processes. They come in different rhythms—alpha, beta, delta, theta and others—and are related to different mental states. Alpha is a relaxation state, beta is a mentally attentive state, delta is a condition of deepest sleep and theta is a state of creativity. Brainwaves also vary with different stimuli such as light, sound and touch.

Patterns of muscle tension, particularly facial muscle tension, are correlated with emotions which may be felt but not expressed. Changes in rapid eye movements also may be associated with emotional feelings and stress or nervousness. Measurements of blood pressure could also be added to our program, as this response is certainly an indicator of the range of comfort to stress.

THE HOLISTIC COMPUTING PROCESS

The user would first need to run a battery of tests on herself in all six of the biofeedback areas. These tests would establish average data points for normal response modes; that is, when the individual is calm or in an otherwise "normal or feeling good" state for that person. The accumulated data would become referent points for future tests run in the daily monitoring program. The user could see where they stand in any given test by comparing their daily score to their standard score (or referent point).

Next, the user would enter known personal health data, for example, high blood pressure, prevalent disorders, susceptibility to stress, etc. With test scores and personal health data together, she could establish a range of acceptable scores for each of the various tests, and a range of unacceptable scores, or scores that indicate poor functioning or undue stress.

Lastly, an assessment of the person's psychoemotional state could be entered, using a sliding scale of 1 to 10, representing negative to positive states. For more precision, each number could identify particular emotional states characteristic of the individual. This, along with the test scores and personal health data, creates the total personal profile.

The user would have to enter information on her activities. There are two categories of activities to be considered: the work-productive and creative activities, and the recreational and relaxation activities. Certain of the activities might rightly fall in either grouping for some individuals, depending upon one's attitude while involved in the activity. Roughly, each grouping would include some of the following: *Work-productive*—Professional activities, office work, writing, studying, physical labor, computer programming, meeting with business colleagues, composing music, housework, doing taxes or bookkeeping. *Relaxation*—Light reading or writing, playing music, playing with kids or pets, making love, playing racquetball, tennis or other favorite sport, doing yoga or taichi, jogging, cashing paycheck and counting money, and indulging in other pleasures.

These activities could be elaborated further, depending upon the user's particular schedule and habits. Each activity or chain or activities could be clearly defined then labeled.

The final data area to be entered into the program is dietary patterns. The information might consist of a combination of what the user knows to be healthy food groups and, within those groups, what the individual particularly likes. Certain kinds of foods that support certain kinds of activities should be grouped together. For example, meats, well-cooked and spicy foods tend to energize, though in a physical rather than mental way. Foods in this group support physical labor, vigorous sports, and social circumstances that require assertiveness. Mental activity is better supported by lighter foods such as fresh fish, fruits, and vegetables. These groups and others could be detailed, then the final groupings defined and labeled.

CORRELATING THE DATA

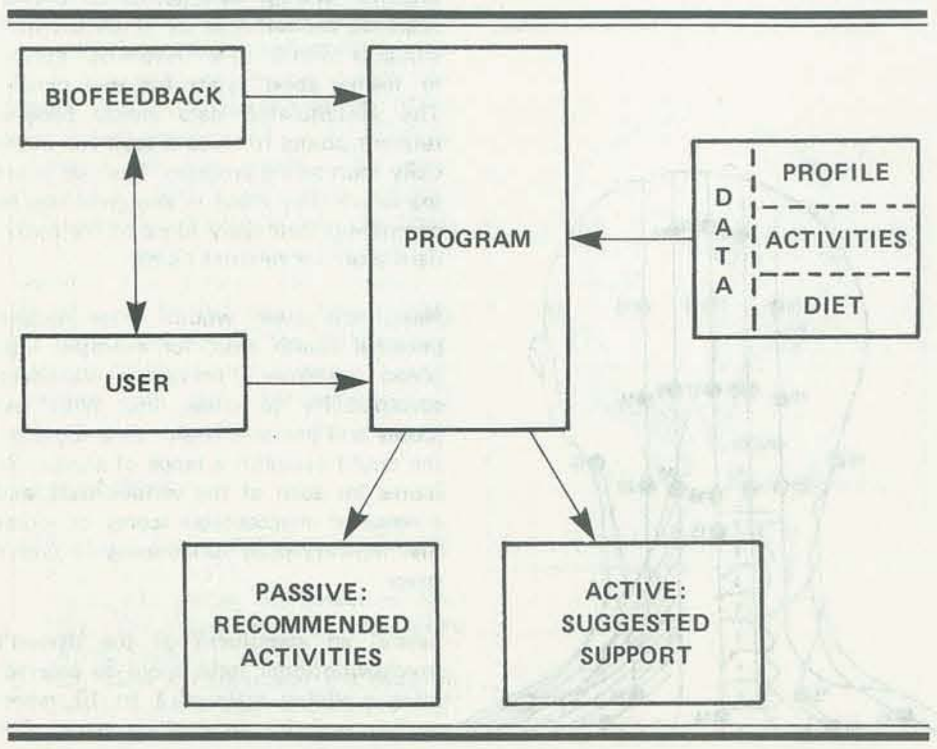
The user can have two different kinds of interaction with this program—passive and active. In the passive mode, the user sits down at the computer, connects with the biofeedback machine, then readings of the physiological response areas are taken. Based upon what the computer knows, it suggests specific activities or a range of activities that are most appropriate for the individual on that day. It also displays relaxation and recreation activities that complement the selected activity, and the dietary pattern that would provide support.

In the active mode, the user still goes through the process of connecting with the biofeedback to give the computer a reading. In addition, though, the user enters a selected activity that she must do on a given day. The computer in turn, using the biofeedback and other data, gives the complementary relaxation activity and diet that would support the user in carrying out the selected activity. In the active mode, the computer helps the user to determine how she can best support herself in doing an activity that she has to do (like conducting a business meeting, studying for an exam or writing an article before deadline).

The details of the program, and particularly the means of connecting groups of data, have yet to be worked out. No doubt the technology for it already exists. Even so, it is still possible to give an example of how the program would work.

If facial muscle tension is registered, blood pressure is a little high, and there is a slight drop in GSR, one can assume that some anxiety exists and that mild stress is evident. With readings like this, the program would indicate a mild exercise such as yoga or taichi, a moderate work-productive activity like office work (if it is in the user's non-stressful category) and letter-writing, and a diet of lightly-cooked foods, fresh fruit, and milk.

Naturally, the program would vary for each individual. With this program, the computer would act as a sort of technological oracle, always truthful and objective, to encourage us toward greater health and productive efficiency.



REFERENCES

1. Ballentine, Rudolph. *Diet & Nutrition, A Holistic Approach.* Pennsylvania: Himalayan International Institute: 1978.
2. Berkeley Holistic Health Center. *The Holistic Health Handbook.* Berkeley: And/Or Press, 1978.
3. Scully, Tim. "Biofeedback and Microcomputers, Parts I and II." *People's Computers,* Vol. 6, No. 2 and 3, 1978.

Adventure

A New Type of Game

Welcome to an astonishing new experience! **ADVENTURE** is one of the most challenging and innovative games available for your personal computer. This is not the average computer game in which you shoot at, chase, or get chased by something, master the game within an hour, and then lose interest. In fact, it may take you more than an hour to score at all, and will probably take days or weeks of playing to get a good score. (There is a provision for saving a game in progress).

This game was inspired by the huge Adventure game which has appeared on large mainframe computers the last several years. But there are important differences. Not only will **ADVENTURE** fit into a relatively small computer, but the 'interpreter' is designed so that different Adventures can be created by changing the data base. So look for more Adventures in the future...

In playing the game you wander thru various 'rooms' (locations), manipulating the objects there to try to find 'treasures'. You may have to defeat an exotic wild animal to get one treasure, or figure out how to get another treasure out of a quicksand bog. You communicate thru two-word commands such as 'go west', 'climb tree', 'throw axe', 'look around'.

Six Different Adventures

ADVENTURELAND (by Scott Adams) - You wander through an enchanted world trying to recover the 13 lost treasures. You'll encounter WILD ANIMALS, MAGICAL BEINGS, and many other perils and puzzles. Can you rescue the BLUE OX from the quicksand? Or find your way out of the maze of pits? Happy Adventuring.....

CS-3007 TRS-80 16K Level II (Machine language)	$14.95
CS-3506 TRS-80 32K Disk (Includes Pirate Adventure)	$24.95
CS-1009 PET 24K (Includes Pirate Adventure. In Basic)	$14.95
CS-5003 SORCERER 16K (Machine language	$14.95
CS-9003 8" CP/M 48K Disk (Includes Pirate Adventure In Microsoft Basic	$24.95

PIRATE ADVENTURE (by Scott Adams) - "Yo Ho Ho and a bottle of rum..." You'll meet up with the pirate and his daffy bird along with many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover LONG JOHN SILVER's lost treasures? Happy sailing matey.....

CS-3008 TRS-80 16K Level II (Machine language)	$14.95
CS-3506 TRS-80 32K Disk (Includes Adventureland)	$24.95
CS-1009 PET 24K (Includes Adventureland. In Basic)	$14.95
CS-5004 SORCERER 16K (Machine language	$14.95
CS-9003 8" CP/M 48K Disk (Includes Adventureland. In Microsoft Basic.	$24.95

ORIGINAL ADVENTURE (by Crowther, Woods, Manning and Roichel) - Somewhere nearby is a collosal cave where others have found fortunes in treasures and gold, but some who have entered have never been seen again. You start at a small brick building which is the wellhouse for a large spring. You must try to find your way into the underground caverns where you'll meet a giant clam, nasty little dwarves, and much more. **This Adventure is Bi-Lingual**—you may play in either **English or French**—a language learning tool beyond comparison. Runs in 32K CP/M system (48K required for SAVE GAME feature). Even includes SAM76 language in which to run the game. The troll says "Good Luck."

CS-9004 8" CP/M 32K Disk	$24.95

MISSION IMPOSSIBLE ADVENTURE (by Scott Adams) - Good Morning, Your mission is to... and so it starts. Will you be able to complete your mission in time? Or is the world's first automated nuclear reactor doomed? This one's well named, its hard, there is no magic but plenty of **suspense**. Good luck.....

CS-3009 TRS-80 16K Level II (Machine language)	$14.95
CS-3507 TRS-80 32K Disk (Includes Voodoo Castle)	$24.95
CS-5005 SORCERER 16K (Machine language	$14.95

VOODOO CASTLE (by Scott Adams) - Count Cristo has had a fiendish curse put on him by his enemies. There he lies, with you his only hope. Will you be able to rescue him or is he forever doomed? Beware the Voodoo Man.....

CS-3010 TRS-80 16K Level II (Machine language)	$14.95
CS-3507 TRS-80 32K Disk (Includes Mission Impossible)	$24.95
CS-5006 SORCERER 16K (Machine language	$14.95

THE COUNT (by Scott Adams) - You wake up in a large brass bed in a castle somewhere in Transylvania. Who are you, what are you doing here, and WHY did the postman deliver a bottle of blood? You'll love this Adventure, in fact, you might say it's LOVE AT FIRST BITE.....

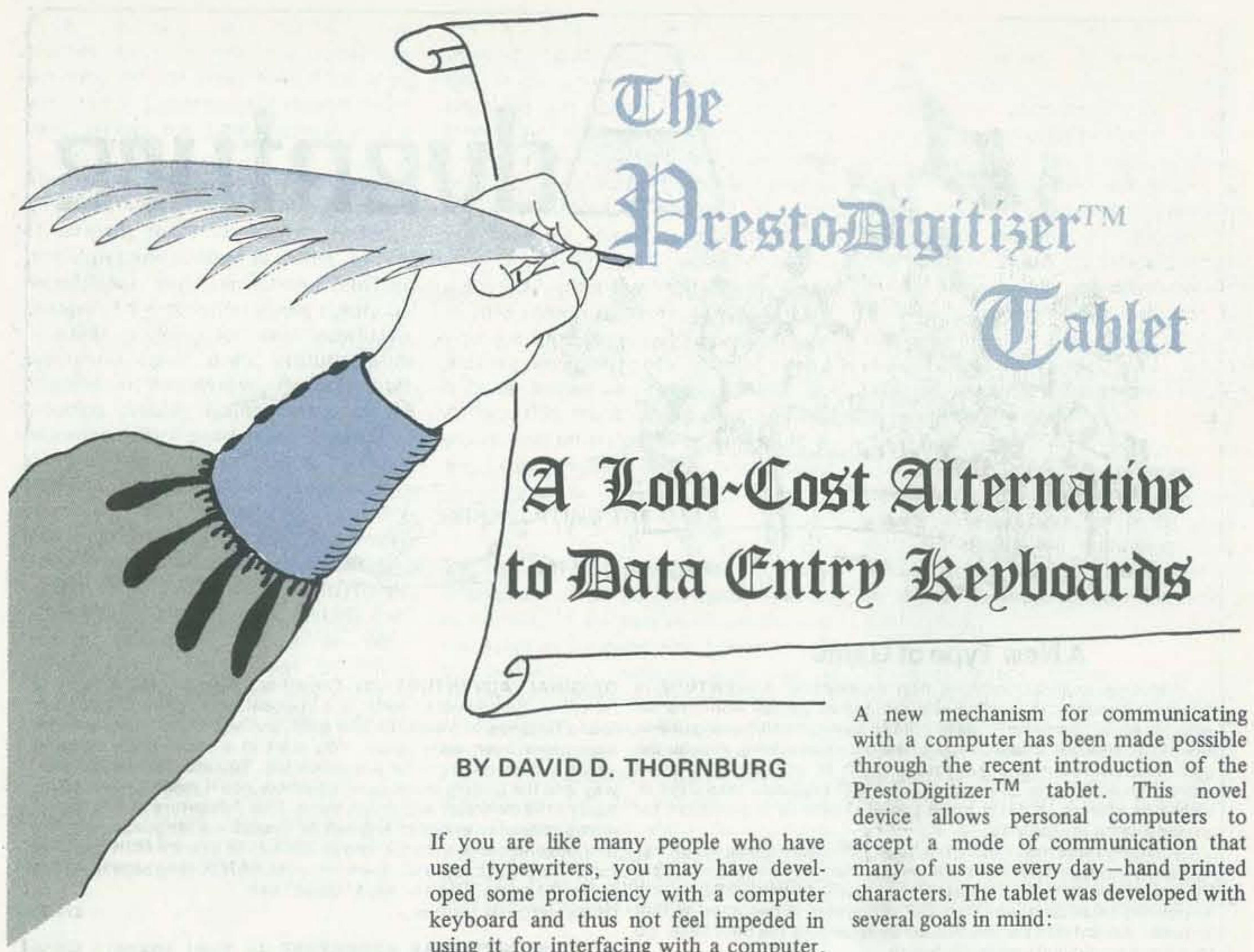
CS-3011 TRS-80 16K Level II (Machine language	$14.95

To order, send payment plus $1.00 shipping or bankcard number to **Creative Computing Software,** P.O. Box 789-M, Morristown, NJ 07960. Or call toll-free **800/631-8112** (In NJ 201/540-0445).

sensational software

creative computing software

Apple II Adventures coming in early 1980!



The PrestoDigitizer™ Tablet

A Low-Cost Alternative to Data Entry Keyboards

"What!? Have my computer understand my handwriting? I can't understand it half the time. You say I can hand print characters and they will be recognized? That the device, software, and instruction manual only cost $50?"

Recently, David has been hearing parts of the above conversation a lot. His company, Innovision, at P.O.Box 1317, Los Altos, CA 94022, has developed a data entry "pad" that accepts hand printed, single characters. It "learns" to read and recognize your printing style.

I have used the pad and know that it is one of those first steps in a long line of innovative consumer products that we all will see in the next few years. In fact, there are simple adaptations of this current device that bubble up once you have tried it–entering patterns of yes/no responses, five "finger" key functions, and so forth. You are reading about the future, and it is now. – *RZ*

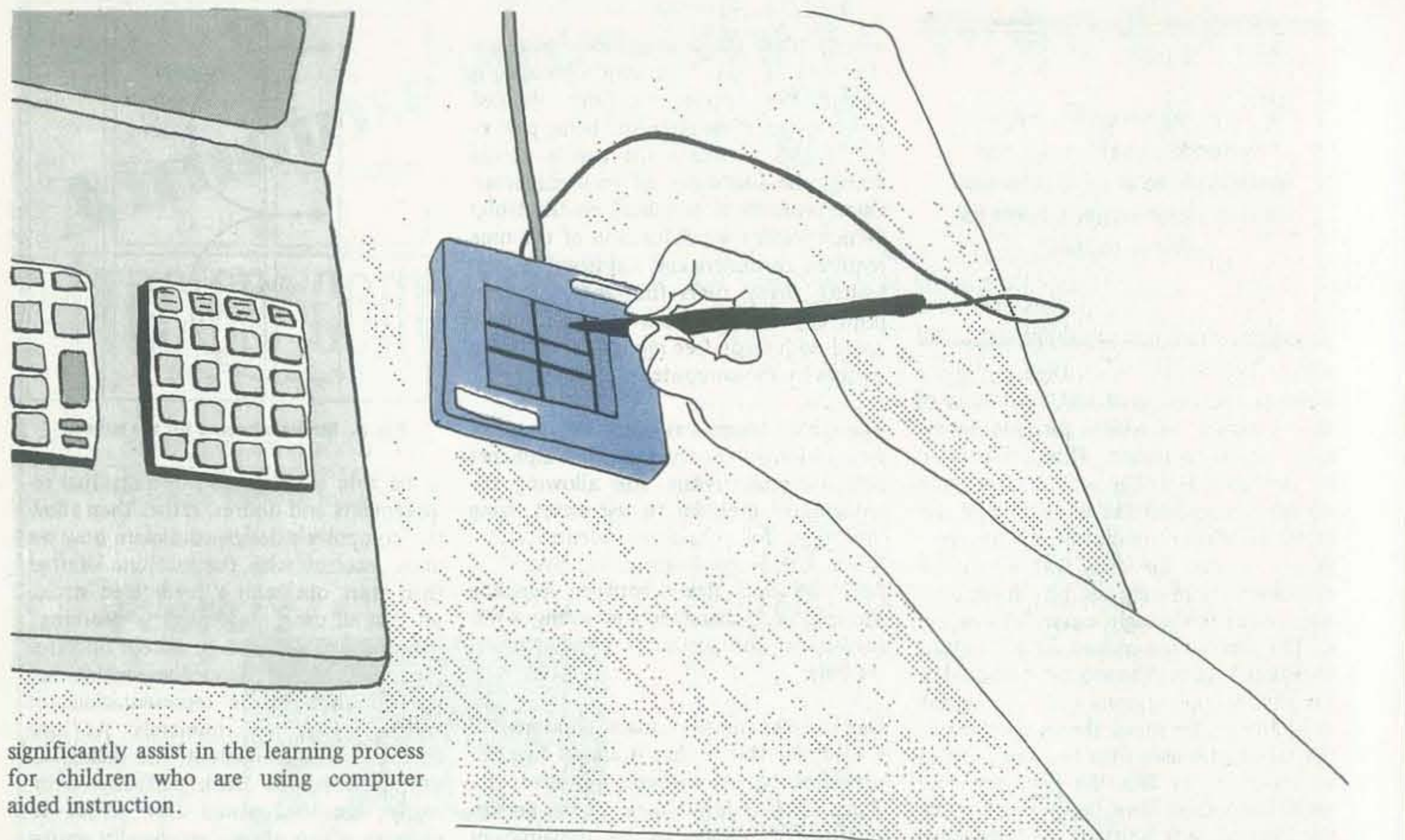
BY DAVID D. THORNBURG

If you are like many people who have used typewriters, you may have developed some proficiency with a computer keyboard and thus not feel impeded in using it for interfacing with a computer. Even the most casual hunt-and-peck typist soon acquires an advantage in this regard when compared to someone for whom the computer keyboard is totally foreign. The problem becomes severe though when the computer is introduced into a classroom of children who are as young as eight or nine. Anyone who has watched children at computer consoles can only marvel at the perseverance with which these enthusiastic youngsters scan the keyboard looking for the right keys to type. If interacting with a computer didn't have other rewards to offset this problem, many people might give up in frustration. Even highly motivated users must lose some sense of continuity as their attention is diverted from answering a question to the task of finding the right keys to push to get their answer into the computer. Clearly, an improved interface between people and computers is needed that allows the computer to accept information in a convenient form for people to use.

A new mechanism for communicating with a computer has been made possible through the recent introduction of the PrestoDigitizer™ tablet. This novel device allows personal computers to accept a mode of communication that many of us use every day–hand printed characters. The tablet was developed with several goals in mind:

- The tablet should allow the computer to adapt to each user's printing style.
- Most users should be able to learn to write on the tablet in a short time (30 minutes or less).
- The tablet should be available for a sufficiently low price so that every personal computer could be equipped with one if desired.

The resulting tablet design (on which a patent is pending) has met all these goals and has provided an additional advantage when it is used in classrooms. When a child is writing an answer to a question, his or her brain is continuing to work on the problem even as the hand is forming a character. It is certainly easier to stop in the middle of a wrong character when you discover a mistake while printing than it is to stop once you have hit a wrong key. This feature alone suggests that handwritten character input might



significantly assist in the learning process for children who are using computer aided instruction.

Most people who have been around large computer installations, or who have attended computer trade shows, are probably familiar with graphics tablets that allow the computer user to draw pictures that are displayed on the computer screen. These graphics tablets, which can cost $500 or more, actually measure the precise location of a special pen on the tablet surface and send these signals, as a pair of coordinates, back to the computer. In contrast to these generalized graphics tablets, the PrestoDigitizer™ tablet is optimized for character recognition and thus is set up to look at stroke direction and sequence rather than to measure precise pen position. This characteristic allows the PrestoDigitizer™ to be sold at one-tenth the price of a low cost generalized graphics tablet, including compact character "learning" and "recognition" software. When used with the 8K Commodore PET, for example, the base software (written in BASIC) occupies only 2K bytes of RAM.

While variations of this tablet are being developed for various applications, the most common configuration is that shown in Figure 1. A frame about 10 cm square contains a circuit board which can be seen through a 6.5 cm square opening. The circuit board is etched into seven distinct regions, six of which are connected to inputs on the parallel user port of the personal computer. Since many manufacturers have a parallel input port for digital joysticks and other add-on peripherals, versions of the PrestoDigitizer™ designed for use with other popular computers will be available at

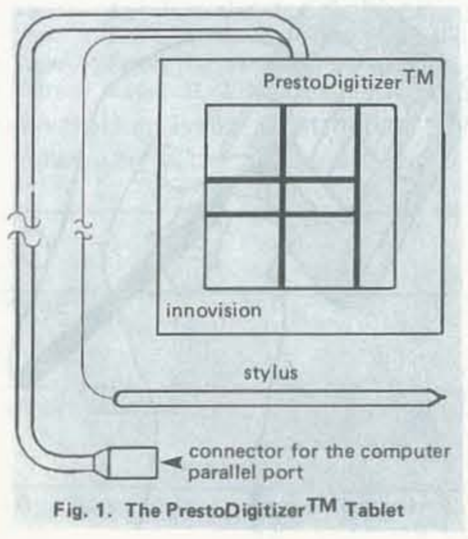


Fig. 1. The PrestoDigitizer™ Tablet

low cost in the near future. Because the tablets presently being shipped are set up to operate with the Commodore PET, this computer will serve as our example in this article. In the PET the parallel I/O port is accessible through an edge connector on the back of the machine. Each of the six tablet regions is connected to a different bit of this port and a special pen which contains an electrical stylus contact is connected to the ground connection on the user port. When the user touches the pen to any of the conductive regions of the tablet surface, the corresponding input of the parallel port is brought from a logical 1 to 0. In this manner, the user is able to have the computer keep track of the region of the tablet in which the pen is located.

In contrast to most small computer peripherals on the market today, the function of the PrestoDigitizer™ tablet is completely controlled by software. For use in character recognition, the program

"An improved interface is needed that allows the computer to accept information in a convenient form for people to use."

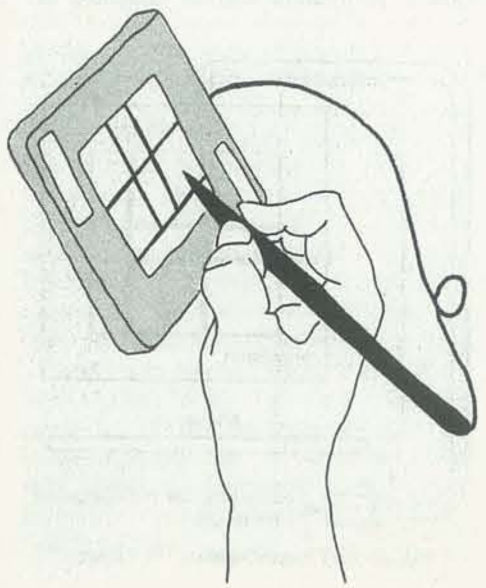
which supports the PrestoDigitizer™ can monitor the user port and make note of the sequence in which the pen moves from region to region. This is illustrated for the letter B in Figure 2. In this figure we have numbered the six regions of the tablet which are monitored by the computer. To draw the letter B as shown, the user starts by placing the pen in region 1 and moves it through region 3 to region 4. The pen is then picked up and moved to region 1 again. The second stroke takes the pen through regions 1, 2, 3, 5, and 4 in sequence. To signal the completion of the letter, the user then touches the pen to region 6 (or lifts the pen from the tablet for a short time, depending on how the program was written). The sequence [13412354] is then associated with one way of writing the letter B.

Once the user has signaled the completion of this character, the program can compare this sequence of strokes against a table of previously stored stroke sequences which are correlated with the letters of the alphabet and with the numerals. If a match is found with one of these stroke sequences, the resulting character will be typed on the screen or treated in the same manner as if it were entered from the keyboard.

New users can train themselves on the PrestoDigitizer™ easily. We have found that it is better to start out with programs which use a limited character set (e.g., the numerals and a few other symbols) before having the operator teach the computer his or her complete repertoire of letters. This allows the computer user to become familiar with the new writing surface, and to see how much variation in the stroke sequence will be tolerated by the program which is performing the recognition task. Usually by the time the third set of strokes has been received by the computer, the accuracy of the recognition program will be nearly perfect. We have found that the recognition accuracy improves if the "learning" process is carried out using carefully defined strokes, with close attention being paid to the regions in which the pen is moved during the formation of each character. Once proficiency is gained on the tablet (which takes a small fraction of the time required to understand a typewriter keyboard), many users find that they can print characters at their normal printing speed with error-free recognition of these strokes by the computer.

The tablet design was optimized to allow recognition of the entire English alphabet and numerals, while still allowing the recognition program to use exact string matching for character identification. Since string comparison in BASIC is fast, the time delay between signaling the end of a character and seeing it displayed on the screen is almost imperceptible.

One of the more valuable features of a tool like this is that it allows adaptive programs to be written. Adaptive programs allow a device such as the PrestoDigitizer™ tablet to be dynamically molded to the style and needs of individual users. While devices of this type are rare, we *should* expect the computer



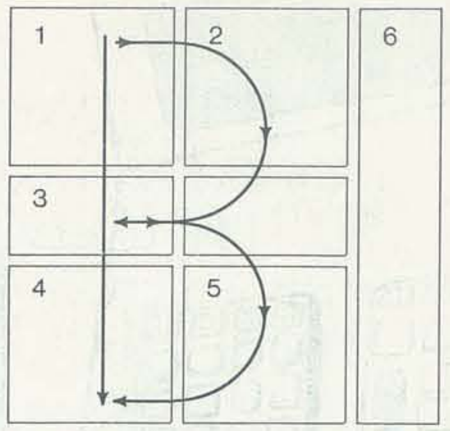
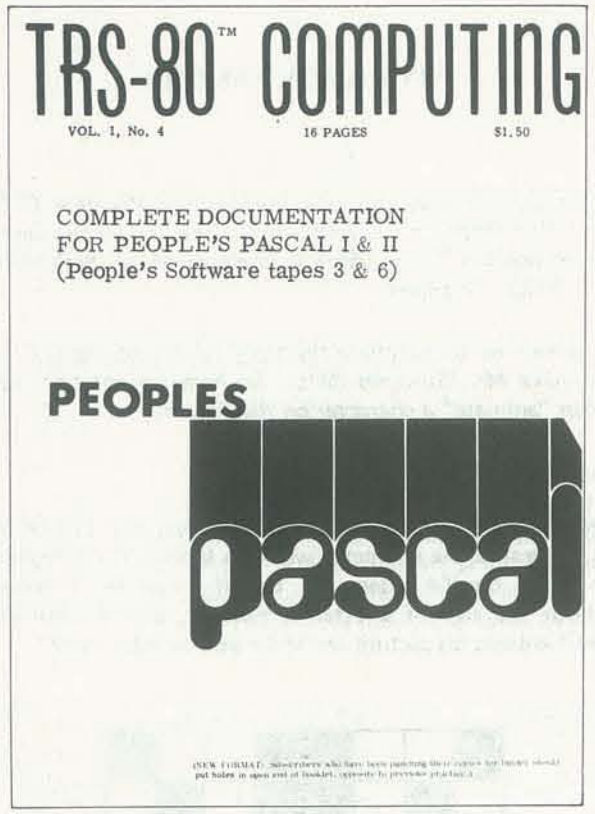


Fig. 2. Stroke sequence for the letter B

to be able to adapt to our individual requirements and desires, rather than allow the computer's design to dictate how we must interact with the machine. Rather than start out with a predefined stroke set that all users must copy, a "learning" program can be used to accept operator generated stroke sequences which are then defined to be representations of various letters and numerals. As long as the user is consistent, the identification process will work perfectly. Ordinarily, the user would draw stroke sequences which appear graphically similar to the desired letter or numeral, but as far as the computer is concerned, the user can make a B that looks like an O and a Q that looks like an L. The correlation between the strokes which are drawn on the tablet and the character that these strokes are chosen to represent is totally up to the person operating the computer. If the user later changes his or her printing style, a character may not be identifiable. At this point the letter which is to correspond to this stroke sequence (Z, for example) can by typed. If this new stroke sequence is encountered in the future, the program will then recognize it as yet another way for drawing the letter Z. From the operator's point of view, the computer starts out with total "ignorance" of his or her printing style. As time goes on the computer gets "smarter," until the computer has fully adapted to the printing style of the individual operating the tablet.

As for the keyboard, novice computer users will have plenty of time to acquire typing skills after gaining expertise with the computer. After all, we each learned to hand print letters on paper before learning how to type. We now have this option with computers as well.

Announcing- TINY PASCAL FOR TRS-80



TRS-80™ COMPUTING

VOL. 1, No. 4 16 PAGES $1.50

COMPLETE DOCUMENTATION
FOR PEOPLE'S PASCAL I & II
(People's Software tapes 3 & 6)

PEOPLES pascal

(NEW FORMAT: subscribers who have been punching their copies for binders should put holes in open end of booklet, opposite to previous practice.)

At last your TRS-80 can run Pascal too! The Chung/Yuen "tiny" Pascal is fully implemented for Level II TRS-80, 16 K and up. You no longer need be left out of the growing group of Pascal users, because People's Pascal gives you everything you need to write structured Pascal programs:
- tiny Pascal compiler
- complete text editor for writing your programs
- complete tiny Pascal monitor
- sample Pascal programs
- user's manual (TRS-80 Computing issue 1:4)

People's Pascal is both a powerful, structured language and "CPU expeditor". People's Pascal programs execute at least four times faster than Basic, and often eight-times faster! Special functions open up the complete graphic capability of TRS-80. You now have the means to write those dazzling, impressive, high-speed graphics programs that are great for games, plotting, statistics, etc.

For the serious computerist, side two of People's Pascal II (tape 6) contains a larger compiler and complete source to the compiler, written in Pascal! This means you can re-compile the compiler, making changes, adding features, etc. (but this will take at least 36 K RAM and a solid knowledge of programming).

With the complete People's Pascal operating system, you can save and load both source (Pascal) programs, and compiled programs, to or from cassette tape. This means that once you have de-bugged a program, you can save the P-code (compiled program) and thereafter, to run the program, you need only load the super-fast P-code.

Here is a partial list of People's Pascal features: recursive procedure/functions • for (loop) • case if/then/else • one-dimensional arrays • write • read constant • repeat/until (loop) • "peek & poke" • plot (graphics for TRS-80)

DEALER INQUERIES INVITED

People's Pascal I (tape 3), is written in Basic, implemented for TRS-80 by John Alexander of Berwick Australia. It compiles more slowly and is harder to use, but includes instructions for converting to disk. People's Pascal II (tape 6) is entirely by the Chung/Yuen team and is a further development stage of their concept. Full documentation included.

Other People's Software tapes $8.

TRS-80 trademark
Tandy Corporation

computer information exchange, inc.
box 158 san luis rey ca 92068

amt.
_______ encl.

PEOPLE'S SOFTWARE, includes 50¢ per tape postage/handling (via 1st class)
( ) Tape I Lev II 34 ass'td programs, or ( ) Lev I, 24, $8; $8.45 in CA
( ) Tape 2 Level II "Common Basic Programs", $8; $8.45 in California
( ) Tape 3 Level II PEOPLE'S PASCAL I, $15.50; $16.40 CA residents
( ) Tape 4 Level I, 17 assorted programs, $8, $8.45 CA residents (tax)
( ) Tape 5 Level II, 28 assorted programs, $8; $8.45 CA residents (tax)
( ) Tape 6 Level II, PEOPLE'S PASCAL II, $23.50; $24.88 CA residents

Make checks payable COMPUTER INFORMATION EXCHANGE

Charge my VISA ( ), MasterCharge ( ) #______________

______________(signed), expires ________

____
name

____
street address

____
city, state, zip

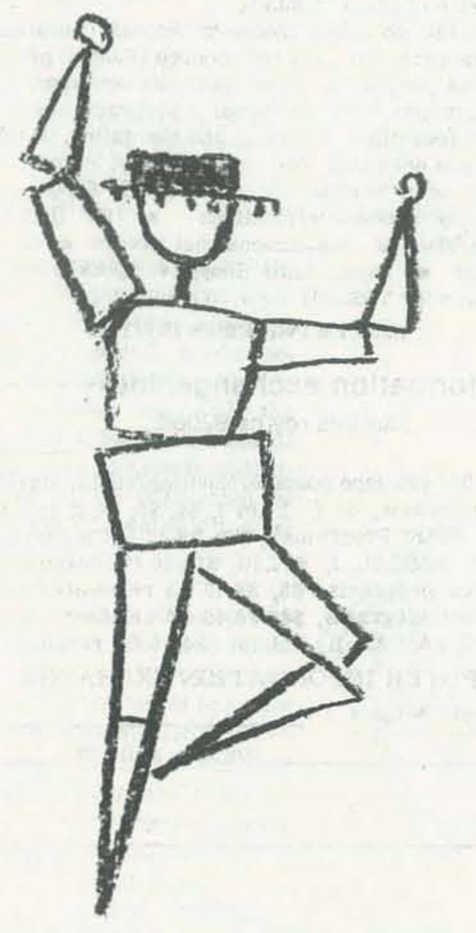
Texas Instruments graphics & animation

Part 2

BY RAMON ZAMORA

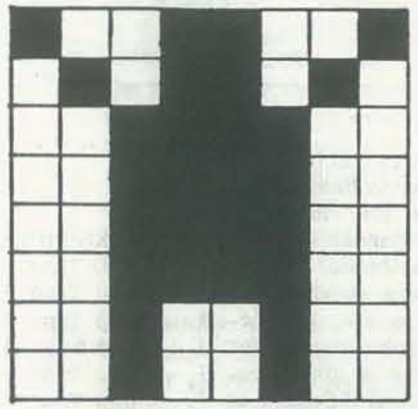
Once again we visit the adventureland of the new TI 99/4 Home Computer. In the first article, Don Inman showed you how to use the TI to create a single graphics character that he called Mr. Bojangles.

Don asked me to complete the story of Mr. Bojangles for you —to make Mr. Bojangles dance. So here we go! Let's see if we can "animate" a character on the TI screen. — *RZ*



MR. BOJANGLES

When we left Mr. Bojangles, he was standing still. The program that generated the character was in a loop that displayed the tiny figure on the video. Mr. Bo (if I can be so familiar) is about the size of a letter or number, and is quite small. We will enlarge his picture and show you how he looks.



You may recall that TI graphics characters are formed by turning on the "dots" within an 8 X 8 area on the screen. The dots are turned on (or left off) when we send eight pairs of the Shorthand Codes to the computer. A detailed discussion of these codes appeared in the last issue of *RC*. (For even more information, look for the book, *Introduction to TI BASIC** from which this series of articles is taken.) For reference, the table of equivalent dot and shorthand codes is repeated below:

* *Introduction to TI BASIC*. Inman, Zamora and Albrecht. Hayden Book Company, Inc., 50 Essex St., Rochelle Park, NJ 07662, 1980.

Table of Equivalent Codes

Dot Code (Binary)	Shorthand Code (Hexadecimal)
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Notice that letters are used for these.

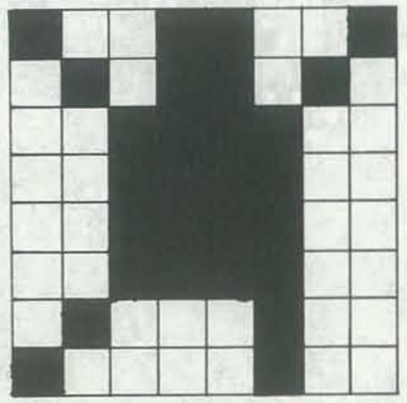
We will return to use this table in a short while.

ANIMATION

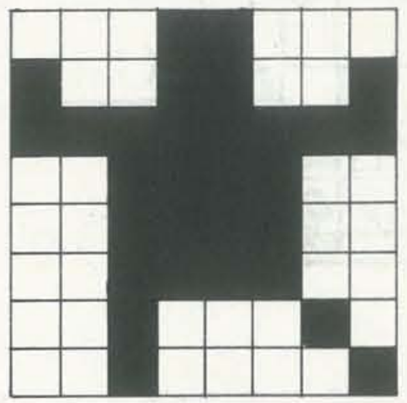
What does it mean to *animate* a character? On the computer screen, if we print a character, wait a short time, clear the screen, wait again, and then repeat the cycle, the character appears to "flash" or "blink." If each time we print the character, we also move its position, then the character appears to "glide" about the screen. There is an illusion, created by the "flashing" and "gliding" that lets us *animate* screen activities.

We must create the illusion of movement to get Mr. Bojangles to dance. We will do this by establishing *two* versions of our original character. Each version will be slightly different so that when we apply the "flashing" technique, Mr. Bojangles will appear to dance.

The original Mr. Bojangles stands rather stiffly. Let's have him change his position by moving one of his legs as follows:



Now we create another version by moving the other leg and both the arms.



If you rapidly alternate your eyes between the two pictures, you get a sense of how Mr. Bojangles looks when he is "dancing."

The TI BASIC program that creates the two characters and animates the dance is given below:

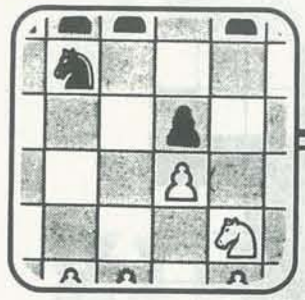
```
 10 REM *** TI ANIMATION ***
 20 CALL CLEAR                   Clear the screen.
 30 A$="995A3C3C3C4484" }        Shorthand codes for two
 40 B$="1899FF3C3C2221" }        characters.
 50 CALL CHAR (96, A$) }         Define the characters as
 60 CALL CHAR (97, B$) }         codes 96 and 97.
 70 CALL COLOR (9, 2, 16)        Set the color.
 80 CALL VCHAR (12, 16, 96)      Display first character.
 90 FOR DELAY=1 TO 100 }         Wait.
100 NEXT DELAY         }
110 CALL VCHAR (12, 16, 97)      Display second character.
120 FOR DELAY=1 TO 100 }         Wait.
130 NEXT DELAY         }
140 GOTO 80                      Make him dance forever.
```

The Shorthand Codes used to create the characters are assigned to string variables (A$ and B$) in lines 30 and 40. Can you decipher how these codes relate to the patterns of dots shown in the figure? Look back at the Table of Equivalent Codes for a hint. If you are stumped, refer to Part 1 of this article in the last issue of *RC*. The new characters are defined to be character codes 96 and 97 in lines 50 and 60. The animation of the dance begins with line 80.

This animation is quite simple. The next steps would be to have Mr. Bojangles dance across the screen, have him turn flips, and perhaps add Ms. Bojangles for some live disco action. With the TI 99/4 Home Computer, your imagination is the only limit that exists.

WHAT'S NEXT

The next issue of *RC* is going to be an issue on Games and Recreations. We will present a unique small game designed for the new TI machine. How about a game that combines sound, color, graphics, and animation? O.K., you got it! See you next issue.

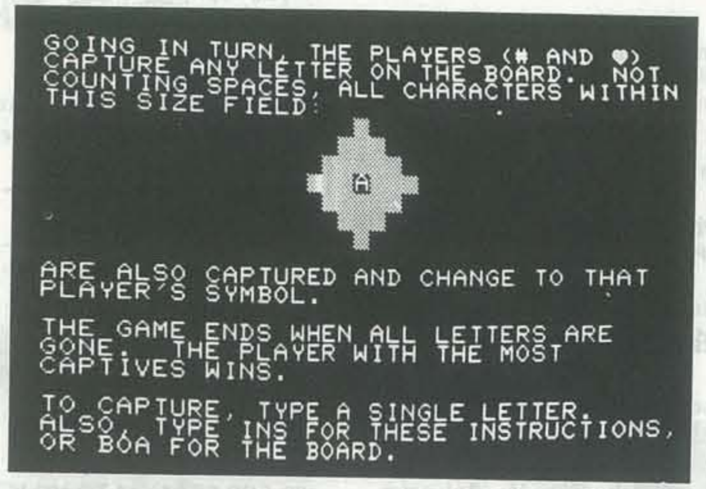


A Game for Your PET

Capture

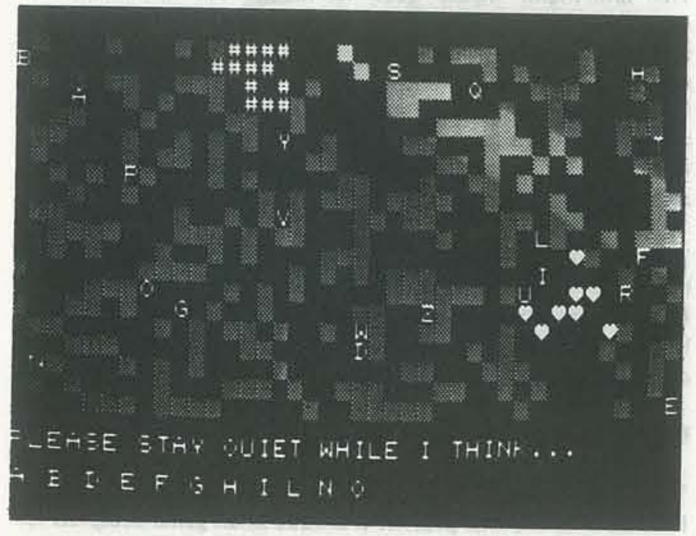
BY MAC OGELSBY

Mac Ogelsby is a prolific and creative computer games designer and teacher. His program makes good use of the PET's graphics. You can play against the computer or with a friend. This program might "capture" a few small people around your house. – RZ



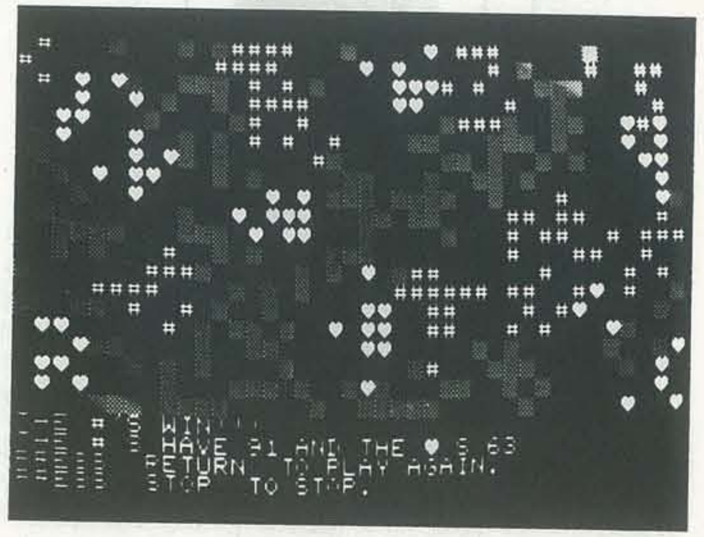
```
GOING IN TURN, THE PLAYERS (# AND ♥)
CAPTURE ANY LETTER ON THE BOARD.  NOT
COUNTING SPACES, ALL CHARACTERS WITHIN
THIS SIZE FIELD:

ARE ALSO CAPTURED AND CHANGE TO THAT
PLAYER'S SYMBOL.

THE GAME ENDS WHEN ALL LETTERS ARE
GONE.  THE PLAYER WITH THE MOST
CAPTIVES WINS.

TO CAPTURE, TYPE A SINGLE LETTER.
ALSO, TYPE  INS  FOR THESE INSTRUCTIONS,
OR  BOA  FOR THE BOARD.
```

Starting the Game



```
PLEASE STAY QUIET WHILE I THINK...
A B D E F G H I L N O
```

Computer Making a Second Move



```
THE # 'S WIN!!!
THE #'S HAVE 81 AND THE ♥'S 63
PRESS  RETURN  TO PLAY AGAIN.
PRESS  STOP  TO STOP.
```

The Computer Wins!

Capture Listing

```
100 REM  NAME:  CAPTURE
105 REM
110 REM  BY:  MAC OGLESBY 2/20/78
115 REM    LAST REVISION 6/29/79
116 REM
120 REM  A GAME FOR 1 OR 2 PLAYERS.
130 REM  COPYRIGHT 1978 BY MAC OGLESBY
140 REM  PERMISSION GRANTED TO USE,
150 REM  BUT NOT TO SELL.
160 REM
1000 REM *** INITIALIZATION
1010 PRINT "[CLR]";: FOR J=1 TO 18: PRINT "[RIGHT]";: NEXT J
1020 PRINT "[S][DOWN][2 LEFT][3 S][DOWN][4 LEFT][5 S][DOWN][6 LEFT]
1021    [7 S][DOWN][6 LEFT][5 S][DOWN][4 LEFT][3 S][DOWN][2 LEFT][S]"
1030 PRINT "[2 DOWN]WELCOME TO[3 DOWN][2 SPACE]C A P T U R E"
1040 PRINT "[4 DOWN]WANT INSTRUCTIONS (YES OR NO)? ";: GOSUB 5000
1050 IF B$<>"YES" THEN 1100
1060 : X=-1
1070 : PRINT "[CLR][10 DOWN]"
1080 : PRINT "I'LL PRINT INSTRUCTIONS AFTER I MAKE"
1090 : PRINT "[2 DOWN]THE PLAYING BOARD..."
1095 : FOR J=1 TO 5000: NEXT J
1100 DIM D$(20),E$(20),L(26),M(26),R(25),C(25),P$(2)
1110 P$(1)="#": P$(2)="[S]"
1140 FOR J=1 TO 25: READ R(J),C(J): NEXT J
1150 DATA -3,0,-2,-1,-2,0,-2,1
1155 DATA -1,-2,-1,-1,-1,0,-1,1,-1,2
1160 DATA 0,-3,0,-2,0,-1,0,0,0,1,0,2,0,3
1165 DATA 1,-2,1,-1,1,0,1,1,1,2
1170 DATA 2,-1,2,0,2,1,3,0
1180 A$="[NULL]": FOR J=1 TO 40: A$=A$+"[SPACE]": NEXT J
1190 FOR J=1 TO 20: D$(J)=A$: NEXT J
1195 PRINT "[CLR]";
1200 FOR J=1 TO 300
1210 : R8=1+INT(20*RND(1)): C8=1+INT(40*RND(1))
1220 : P8=32767+C8+40*(R8-1): IF PEEK(P8)<>32 THEN 1210
1230 : C$="[&]": IF J>26 THEN POKE P8,102
1240 : IF J<27 THEN L(J)=100*R8+C8: C$=CHR$(J+64): POKE P8,J: M(J)=L(J)
1250 : GOSUB 6000
1260 NEXT J
1300 FOR J=1 TO 20: E$(J)=D$(J): NEXT J
1400 IF X<0 THEN GOSUB 9000
2000 T=2
2100 PRINT "[HOME]";: FOR J=1 TO 21: PRINT "[DOWN]";: NEXT J
2110 PRINT: PRINT "HOW MANY HUMAN PLAYERS (1 OR 2)? ";
2120 GOSUB 5000: IF B$="[NULL]" THEN 2120
2130 IF B$<>"1" AND B$<>"2" THEN 2110
2140 PL=VAL(B$)
2150 IF PL=2 THEN GOSUB 7000: GOTO 3000
2200 PRINT "[CLR][6 DOWN][6 RIGHT]OK, I'LL PLAY THE #'S"
2210 PRINT "[3 DOWN]WHO GOES FIRST (1=COMPUTER  2=YOU)? ";
2220 GOSUB 5000: IF B$="[NULL]" THEN 2220
2230 IF B$<>"1" AND B$<>"2" THEN PRINT: GOTO 2210
2240 T=3-VAL(B$): GOSUB 7000
3000 REM *** GENERATE COMPUTER'S MOVE
3100 T=3-T
3110 IF PL=2 OR T=2 THEN 4000
3130 PRINT "PLEASE STAY QUIET WHILE I THINK...": E=0: PRINT
3160 FOR J=1 TO 26
3200 : IF L(J)=0 THEN 3900
3210 : PRINT CHR$(J+64);"[SPACE]";
3220 : R=INT(L(J)/100): C=L(J)-R*100
3240 : Q=0
3250 : FOR K=1 TO 25
3260 : : R1=R+R(K): IF R1>20 OR R1<1 THEN 3400
3270 : : C1=C+C(K): IF C1>40 OR C1<1 THEN 3400
3280 : : X$=MID$(D$(R1),C1,1): IF X$="[SPACE]" OR X$="#" THEN 3400
3290 : : Q=Q+1
3400 : NEXT K
3500 : IF Q>E THEN M$=CHR$(J+64): E=Q: GOTO 3900
3510 : IF Q=E THEN M$=M$+CHR$(J+64)
3900 NEXT J
3910 X=1+INT(LEN(M$)*RND(1))
3920 Z$=MID$(M$,X,1): X=ASC(Z$)-64
3950 PRINT "[CLR][10 DOWN][12 RIGHT]I CAPTURE  ";Z$: GOTO 4300
4000 REM *** HUMAN'S MOVE
4110 PRINT "THE ";P$(T);"'S CAPTURE...";
4200 GOSUB 5000: IF B$="[NULL]" THEN 4200
4210 IF B$="INS" THEN GOSUB 9000: GOTO 4110
4220 IF B$="BOA" THEN GOSUB 7000: GOTO 4110
4230 IF LEN(B$)<>1 THEN 4260
4240 X=ASC(B$)-64: IF X<1 OR X>26 THEN 4260
4250 IF L(X)<>0 THEN 4300
4255 PRINT: PRINT "THAT LETTER HAS BEEN CAPTURED.": GOTO 4270
4260 PRINT: PRINT "I DON'T UNDERSTAND YOUR MOVE..."
4270 PRINT "  TYPE  INS  FOR INSTRUCTIONS"
4280 PRINT "  TYPE  BOA  TO PRINT THE BOARD"
4290 GOTO 4110
4300 R=INT(L(X)/100): C=L(X)-R*100
4390 C$=P$(T)
4400 FOR J=1 TO 25
4410 : R8=R+R(J): IF R8<1 OR R8>20 THEN 4500
4420 : C8=C+C(J): IF C8<1 OR C8>40 THEN 4500
4430 : Z$=MID$(D$(R8),C8,1): IF Z$="[SPACE]" THEN 4500
4440 : X=ASC(Z$): IF X>64 AND X<91 THEN L(X-64)=0
4450 : GOSUB 6000
4500 NEXT J
4520 GOSUB 7000: X=0
4600 FOR J=1 TO 26: X=X+L(J): NEXT J
4610 IF X=0 THEN 8000
4700 GOTO 3000
5000 REM *** GET INPUT
5100 B$="[NULL]"
5120 GET A$: IF A$<>"[NULL]" THEN 5120
5200 GET A$: IF A$="[NULL]" THEN 5200
5300 IF ASC(A$)=13 THEN RETURN
5400 B$=B$+A$: PRINT A$: GOTO 5200
6000 REM *** UPDATE BOARD
6100 IF C8=1 THEN D$(R8)=C$+RIGHT$(D$(R8),39): RETURN
6200 IF C8=40 THEN D$(R8)=LEFT$(D$(R8),39)+C$: RETURN
6300 D$(R8)=LEFT$(D$(R8),C8-1)+C$+RIGHT$(D$(R8),40-C8): RETURN
7000 REM *** PRINT BOARD
7010 PRINT "[CLR]";
7100 FOR J=1 TO 20: PRINT D$(J);: NEXT J
7200 PRINT: RETURN
8000 REM *** END OF GAME
8100 PRINT "[CLR]NOW, LET'S SEE WHO WON..."
8200 H1=0: H2=0
8300 FOR J=1 TO 20
8310 : PRINT D$(J);
8320 : FOR K=1 TO 40
8330 : : A$=MID$(D$(J),K,1)
8340 : : IF A$="#" THEN H1=H1+1: GOTO 8400
8350 : : IF A$="[S]" THEN H2=H2+1
8400 : NEXT K
8500 NEXT J
8600 IF H1<>H2 THEN 8700
8620 PRINT "TIE GAME!!! EACH HAS";H1;"CAPTIVES.": GOTO 8720
8700 PRINT "THE ";P$((SGN(H2-H1)+3)/2);"'S WIN!!!"
8710 PRINT "THE #'S HAVE";H1;"AND THE [S]'S";H2
8720 PRINT "PRESS  RETURN  TO PLAY AGAIN."
8730 PRINT "PRESS  STOP  TO STOP."
8740 GOSUB 5000
8750 IF LEN(B$)>0 THEN END
8760 PRINT "[CLR][8 DOWN][8 RIGHT]1 = SAME SETUP"
8770 PRINT "[DOWN][8 RIGHT]2 = NEW BOARD[2 DOWN]"
8780 PRINT "PLEASE TYPE 1 OR 2... ";
8790 GOSUB 5000: IF B$="[NULL]" THEN 8790
8800 IF B$="2" THEN RUN 1100
8810 IF B$<>"1" THEN PRINT: GOTO 8780
8820 FOR J=1 TO 20: D$(J)=E$(J): NEXT J
8830 FOR J=1 TO 26: L(J)=M(J): NEXT J
8840 GOSUB 7000: GOTO 2000
9000 REM *** INSTRUCTIONS
9010 PRINT "[CLR]";
9100 PRINT "GOING IN TURN, THE PLAYERS (# AND [S])"
9110 PRINT "CAPTURE ANY LETTER ON THE BOARD.  NOT"
9120 PRINT "COUNTING SPACES, ALL CHARACTERS WITHIN"
9125 PRINT "THIS SIZE FIELD:";
9130 PRINT "[2 SPACE][DOWN][&][DOWN][2 LEFT][3 &][DOWN][4 LEFT][5 &]
9131    [DOWN][6 LEFT][3 &]A[3 &][DOWN][6 LEFT][5 &][DOWN][4 LEFT]
9132    [3 &][DOWN][2 LEFT][&]"
9140 PRINT "[DOWN]ARE ALSO CAPTURED AND CHANGE TO THAT"
9150 PRINT "PLAYER'S SYMBOL."
9160 PRINT
9170 PRINT "THE GAME ENDS WHEN ALL LETTERS ARE"
9190 PRINT "GONE.  THE PLAYER WITH THE MOST"
9200 PRINT "CAPTIVES WINS.": PRINT
9220 PRINT "TO CAPTURE, TYPE A SINGLE LETTER AND"
9230 PRINT "PRESS RETURN.  TYPE  INS  FOR THESE"
9240 PRINT "INSTRUCTIONS, OR  BOA  FOR THE BOARD."
9300 PRINT: PRINT: PRINT "PRESS RETURN WHEN YOU'RE READY...";
9400 GOSUB 5000
9410 GOSUB 7000: RETURN
```

Note [1].– Line 1021 is continuation of 1020; lines 9131 and 9132 are continuations of 9130.

Note [2].– If the program is entered with all spaces and comments as shown, an 8K PET may not have enough room to run without peculiar errors. Discard unnecessary spaces and comments, and if needed, omit line 1300 (which will remove the possibility of replay with same board).

Note [3].– [ ] usually means shift, then type what's enclosed. Sometimes the [ ] enclose a description of a non-printing character. "[3 &]" means hold down shift and type 3 ampersands.

GAMES TO PROGRAM

Some Simple Language Games and Partial Dictionaries

BY HERBERT KOHL

Herb and his wife Judith are co-directors of Coastal Ridge Research and Education Center in Pt. Arena, CA. The center is a non-profit educational organization designed to provide both theoretical and practical support to people involved in progressive educational activities.

In this ongoing series of articles, Herb challenges you to send in programs that implement these game ideas. Herb, an educator and writer, promises to keep supplying you with a stream of educational (and fun!!) games for you to program.
– RZ

In addition to writing *Alice in Wonderland* and *Through the Looking Glass*, Lewis Carroll was a master at creating games and puzzles. Here is a word puzzle called *Doublets* which he created in 1879. It is easy to understand the puzzle by considering two simple examples that Carroll uses:

- Drive *pig* to *sty*
- Raise *four* to *five*

The answers to these challenges are:

- pig-wig-wag-way-say-sty
- four-foul-fool-foot-fort-fore-fire-five

In each case the transformation is made by changing one letter in each step and making a new word with it until the goal is reached. The rule is that at each step a word must be made. No nonsense syllables are allowed. Here are a number of doublets:

- Turn *eye* to *lid*
- Make *tea hot*
- Change *elm* to *oak*
- Put *rouge* on *cheek*

Answers:

- eye-dye-die-did-lid
- tea-sea-set-sot-hot
- elm-ell-all-ail-air-fir-far-oar-oak
- rouge-rough-sough-south-sooth-booth-boots-boats-brats-brass-crass-cress-crest-chest-cheat-cheap-cheek

There are a number of interesting questions that can be asked about doublets. Can any three letter word be turned into any other one using Carroll's rules for transforming words? What about four letter words? In order to come up with answers for these questions partial dictionaries of English are needed, one a dictionary of acceptable three letter words and another of four letter words. These dictionaries would be very useful for playing other word games. For example it is possible to extend Carroll's rules so that words can be replaced by ones that rhyme with them, so that letters can be inverted, so that anagrams can be allowed, and so forth. At one point it might be possible to develop nontrivial rules so that all three and four letter words are transposable to each other.

Three letter words can also be used to develop magic word squares–grids where the words read the same across and down.

a	t	e
t	a	g
e	g	g

a	n	t
n	e	w
t	w	o

f	l	y
l	e	e
y	e	e

It would be interesting to see if some word squares can be transposed into others using simple rules of letter substitution and inversion. Again dictionaries of acceptable words could be used either to actually find words to solve these problems or to correct solutions.

A Learning Program for Problem Readers

ERIC SEVCIK AND JIM SEVCIK

In the May-June issue of this year, we published an article, "Peter Can Now Read," about a dyslexic boy whose reading problem was helped by a word timer program on the TRS-80. The reader response was enormous.

Eric Sevcik, a 16 year-old dyslexic lad, began with the basic idea in that article and developed a PET equivalent to the reading program. Our cheers to Eric for a notable accomplishment!

The program tape is being sold through Computer Pathways Unlimited, 2151 Davcor Street SE, Salem, OR 97302. – TD

BACKGROUND (Jim Sevcik)

Eric Sevcik was diagnosed as dyslexic when he was in kindergarten. Primary grades were a constant frustration for Eric because of his inability to read or write. With poor fine muscle control and dyslexic perception, writing, spelling and reading were almost impossible. These problems caused Eric's teachers to discount his superior reasoning skills and knowledge of science. During these years, Eric's self image was under steady attack.

By sixth grade, Eric's reading progress was improving. Oral reports were often an acceptable substitute for written work. Seventh grade, or Junior High, was a good year for Eric with only handwriting and spelling problems. He had continued to excel in science.

Diabetes struck hard in April of Eric's 8th-grade year. In and out of school and hospital was the story for him during the next 9 months. With the strong positive support of his teachers and school, Eric completed all 9th grade work and was credited.

Eric entered High School with positive expectations. Because Eric was "different," the teachers and some staff inadvertently attempted to destroy Eric's self-worth with frequent comments like, "Sick kids with your handwriting don't belong in school."

Adjusting to a diabetic situation, a new school, adolescence and an aggressive, defeatist school environment was too much all at once. Two thirds of Eric's sophomore year were lost. His spirit sank, but didn't quite drown.

During this time Eric often retreated to his room to find inner strength. It was then that he started to read about electronics and computers. Last year, by some great fortune, a PET 8K computer was donated to Eric's elementary school by an interested parent. In a visit to his old school, Eric expressed interest in the PET.

In the "Peter Can Now Read" article in *RC* Eric saw a possible way to help the PET break into the elementary classroom. During the summer, the donor arranged to loan the computer to Eric for two weeks if he would try to convert the TRS-80 program for the PET. (Everyone thought that BASIC-to-BASIC conversion would be a relatively uncomplicated task.)

Finally, the *school* had confidence and trust in Eric! The school was offering him a private computer in exchange for a simple programming challenge!

Eric's family teamed up to give him maximum freedom and support. From 7 a.m. to 11 p.m. every day, stopping only for biological needs, Eric worked on the program conversion. With no one to turn to for technical help, he relied primarily on the PET User Manual and Radio Shack reference books. Eric knew that elementary school kids needed this program.

The task was completed! As of the end of those two weeks, Eric walks straighter, talks clearer and laughs more. This year, high school is no threat. His handwriting and spelling still need work, but he has started to type. Most important to Eric–the kids in his old elementary school now have access to infinitely patient reading help in the PET program.

INTRODUCTION TO PET READING PROGRAM
(Eric Sevcik)

This program is the PET equivalent of the "Word Timer Program" listed in the article "Peter Can Now Read." The PET Reading Program does everything the original program does, and it uses special PET graphic characters. Additionally, several different options have been built into the PET program that are not in the original TRS-80 program.

I wrote this program because I had never before been given an opportunity to write a program for someone else, and it was the first time I had been given a problem to solve on a computer.

CHANGING THE WORD LIST AND SCAN RATE

The list of words that the program uses is contained in the data statements 3000-3440. The words are organized into sections of ten words each, though the section length can be changed for convenience.

The words for each section are in the first line of the paired data statements. This line contains the Group Number, then the Section Number followed by the words separated by commas. The second line of the paired data statement contains the code for controlling the individual scan rate (speed of letter highlights) for each word.

To replace words, just recode the first data statement with the new replacement words. Be sure that the second statement has the same number of scan rate codes as the words in the first statement. In changing the scan rate, remember that the larger the number the slower the speed. I suggest that you change one pair of data statements and test them successfully, before embarking on a wholesale change.

Materials Needed to Run Program

- PET Computer
- Pupil (someone to interact with the program)
- Teacher (someone to set up and start the program)
- PET Reading Program tape
- PET Reading Program Word tape (for voice reinforcement)
- External tape recorder connected to PET as prescribed

Note: The last two items are not needed if the voice reinforcement feature is not being used.

EXTERNAL TAPE RECORDER

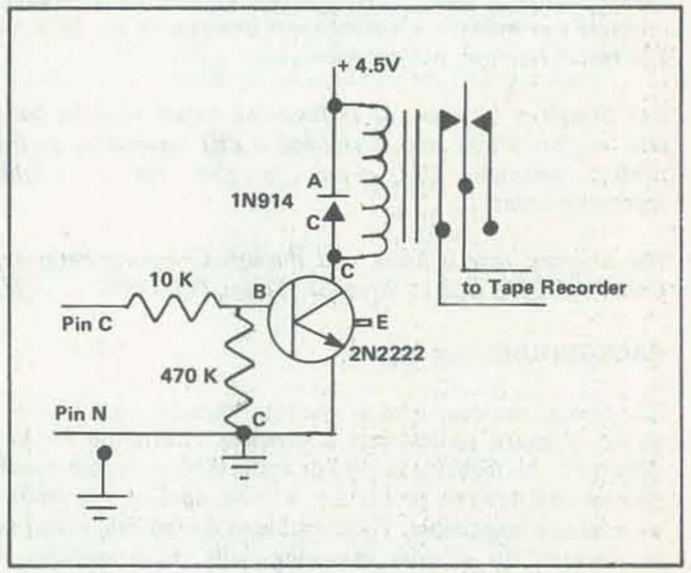
A cassette recorder is needed to load the program into the computer, of course. Two options are possible:

A. Purchase the Pet accessory and follow directions for hookup, then modify the POKE commands as necessary.
B. Purchase an inexpensive portable recorder and connect it through a switching interface. (Option B is about ½ the price of Option A.)

The interface allows the PET to control the power supply to the tape recorder. Construction and testing is best done on the P-board. Once tested, you can shorten the leads, retest, then liberally apply epoxy glue. The assemblage is now a solid unit that will take rough handling.

Parts List

Experimenter Socket (small P-block)
10k OHM Resistor ¼ watt
470k OHM Resistor ¼ watt
2N2222 NPN 500 MA HFE: 100-300 Transistor
1N914 Switching Diode
5VDS DPDT DIP Relay



Recording a New Voice Tape of Words

- Be sure the external tape recorder is attached as if you were going to run the program.
- Insert a blank tape into the external tape recorder and rewind fully.
- RUN the Reading Program except when you are asked to press PLAY, press both the PLAY and RECORD controls.
- Answer NO to questions and the options to make a new tape.
- Plug the microphone into the external recorder.
- Hit D and follow the instructions displayed on the PET.

Program Options

- DO YOU WISH TO START AT THE BEGINNING OF THE WORDS?
  YES will start the word list at the first word. *Warning:* the voice tape will *not* be synchronized unless you stop the program and start from the beginning.
  NO will let you go to the next option.
- DO YOU WANT TO GO TO THE NEXT SECTION?
  YES will advance the program and voice tape to the next section of words.
  NO will let you go to the next option.
- DO YOU WANT TO CHANGE PACE?
  YES will bring you back to the ? PACE CHANGE ? request.
  NO will let you go to the next option.
- DO YOU WANT TO LOOK THROUGH THE SECTION OF WORDS?
  YES will rapidly present the next section's words.
  NO will let you go to the next option.
- DO YOU WANT TO MAKE A TAPE FOR THE COMPUTER?
  YES refer to the instructions titled "Recording a New Voice Tape of Words."
  NO will let you go to the next option.
- DO YOU WANT TO VERIFY TAPE WORD SYNC?
  YES will display each word and play the voice tape, thus allowing you to check the synchronization. *Warning:* a sync check requires that none of the other options have been executed and that you are starting at the very beginning.
  NO will cause the PET to display BYE and end the program.

Instructions to Run Program

Loading the program
- Turn on PET
- Insert the PET Reading Program tape, program side up and rewind.
- Type LOAD READING and hit RETURN key.
- Press PLAY control when PET tells you to do so.
- Shortly the PET will display FOUND READING, then LOADING.
- Finally the PET will display READY.

Running the Program
- If using voice reinforcement, insert word tape into the external tape recorder.
- Type RUN and hit the RETURN key.
- The program will list a series of instructions to the teacher. Hit the D key when each is done. (If not using voice reinforcement, ignore the instructions and hit D for the first 3 instructions anyway.)

PET Instructions
- MAKE SURE THE INTERFACE IS HOOKED TO THE COMPUTER. This is to insure that you check the interface connections.
- REWIND THE TAPE ALL THE WAY. Verify that the word tape is inserted correct side up and is rewound all the way in the external tape recorder.
- PRESS PLAY ON TAPE. Press PLAY control on the external tape recorder.
- The PET will display I AM NOW MOVING TO THE BEGINNING OF THE RECORDING. At this time the word tape is being positioned by the program. *Do Not Touch Anything!!*
- The PET will display ? PACE CHANGE ? Follow the instructions to speed up or slow down the word presentations.
- The section and group number identity of the next set of words to be presented will be displayed.
- The PET will ask you if you wish to go on with the indicated set of words.
  A YES response will cause the program to display the words.
  A NO response will allow you to select from the available options.
  (See Program Options, shown above.)
- The screen will clear then display the section, group and word numbers, before presenting the word letter by letter.
- The letter highlight sequence will start after all the letters of the word are displayed.
- The pupil is now to try to read then pronounce the word displayed.
- Hitting the minus (−) key will cause the program to pause the highlights so that the teacher can work with the pupil if desired.
- After the pupil has decoded and pronounced the word hit the asterisk (*) key to stop the highlights and to activate the word (voice reinforcement) tape. (If not using voice reinforcement hit the * anyway.)
- The PET will then display DID YOU READ THE WORD RIGHT? PLEASE TYPE 1 FOR YES OR 0 FOR NO. Follow the directions.
- The next word of the Section will appear after either a 1 or 0 is hit.
- A rocket launch display will show the correct number as each section is ended.
- At this time, the average time used per word will be displayed in the upper right corner.
- The program will recycle from the ? PACE CHANGE ? request until the word list is exhausted or until you indicate that you do not want to go on by typing in NO to the continue request.

Listing

```
READY.

 1 REM **********
 2 REM *READING *
 3 REM *  BY    *
 4 REM * ERIC W.*
 5 REM *SEVCIK  *
 6 REM *        *
 7 REM * SALEM  *
 8 REM *  ORE.  *
 9 REM *JULY1979*
 10 REM **********
 14 P(3)=59471
 15 I=0:POKE59459,1:POKEP(3),0:GOTO1000
 24 FORI=1TOW
 25 Z=32:PRINT"■WORD # ";I,,"SECTION # ";S(2),,"GROUP # "G(1):GOTO100
 40 FIX(2)=TI-FIX(1):FIX(3)=FIX(2)+FIX(3):GOSUB440
 50 GOTO700
 55 NEXTI
 60 GOTO800
 70 GOTO1045
 100 REMFORM WORD IN MIDDLE OF SCREEN---
 110 L=LEN(W$(I)):KID=100+VAN
 111 FIX(1)=TI
 120 FORX=1TOL:A$=LEFT$(W$(I),X):PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■";A$:T=KID
 130 GOSUB600:NEXTX:T=200:GOSUB600:T=10+PET:GOSUB600
 300 A=83:B=30:P(1)=33220:P(2)=33300:REM MROCRO BLOT DRIVER---
 310 FORX=0TOL-1:POKEP(1)+X,A:POKEP(2)+X,B:T=D(I)+PET:GOSUB600:GETI$
 326 IFI$="-"THENGOSUB631
 330 NEXTX
 350 PRINT"■■■■■■■■■■■■■■"
 360 PRINT"■■■■■■■■■■■■■■■"
 370 GOSUB600:GETI$
 373 IFI$="*"THEN40
 375 IFI$="-"THENGOSUB631
 380 GOTO310
 440 G(0)=200:GOSUB480:RETURN:REM PLAY--WORD---
 480 POKEP(3),1
 482 G(2)=TI+G(0)
 484 IFTI>=G(2)THEN490
 486 GOTO484
 490 POKEP(3),0:RETURN
 500 RESTORE:GOTO504:REM RECORD TAPE---
 504 PRINT"■GET INS.■OUT OF PAMPHLET"
 515 PRINT"■■■■      HIT (D) WHEN DONE READING INS.":GOSUB1002
 519 INPUT"HOW MANY SECTIONS OF WORDS";S:FORX=1TOS:GOSUB640:FORI=1TOW
 530 PRINT"■WORD # ";I,,"SECTION # ";S(2),,"GROUP # "G(1)
 535 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■";W$(I):T=270:GOSUB600:PRINT"■■5":T=35:GOSUB6
00
 542 PRINT"■■■■4":GOSUB600:PRINT"■■■■■■3":GOSUB600:PRINT"■■■■■■■■2":GOSUB600
 548 PRINT"■■■■■■■■■■1":GOSUB600:PRINT"■■■■■■■■■■■■■■START":G(0)=201:GOSUB480
 555 NEXTI,X
 570 PRINT"■END OF WORDS":END
 600 REM DLEY ***
 610 T(1)=TI+T
 620 IFTI>=T(1)THENRETURN
 627 GOTO620
 631 REM PAUSE******
 632 U(1)=T:T=250:GOSUB634
 633 T=U(1):GOTO620
 634 T(2)=TI+T
 635 IFTI>=T(2)THENRETURN
 636 GOTO635
 640 REM GET  DATA *********
 650 READG(1):READS(2):V=G(1):P=I:J=S(2)
 660 READW
 670 FORI=1TOW:READW$(I):NEXTI
 680 FORI=1TOW:READ D(I):NEXTI
 690 RETURN
 700 PRINT"DID YOU READ THE WORD RIGHT?■■■■■  ■■■■■HIT (1) FOR YES OR (0) FOR NO
 704 GETI$:IFI$=""THEN704
 710 IFI$="1"THENR=R+1
 711 IFI$="0"ORI$="1"THENGOTO55
 720 PRINT"PLESE TYPE A (1) FOR YES OR (0) FOR NO":GOTO704
 725 REM SERCH MODE****
 730 I=I+1:IFI>WTHENI=1
 735 PRINT"■WORD # ";I,,"SECTION # ";S(2),,"GROUP # "G(1)
 740 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■";W$(I):T=100:GOSUB600
 745 PRINT"■■■■■■HIT ANY TO CONTINUE.HIT(S) TO STOP"
 746 GETA$:IFA$=""THEN746
 750 IFA$="S"THENGOTO1050
 755 GOTO730
 800 PRINT"■":REM ROCKET SCORE*****
 805 PRINTTAB(25)"SEC/WORD ";INT(FIX(3)/60)/W
 806 FIX(1)=0:FIX(2)=FIX(1):FIX(3)=FIX(2):PRINT"■"
 809 IFR>10THENR=10
 810 PRINT"■10":PRINT"■  ":PRINT"■ 9":PRINT"■  ":PRINT"■ 8":PRINT"■  ":PRINT"■ 7
"
 811 PRINT"■  ":PRINT"■ 6":PRINT"■  ":PRINT"■ 5":PRINT"■  ":PRINT"■ 4"
 812 PRINT"■  ":PRINT"■ 3":PRINT"■  ":PRINT"■ 2":PRINT"■  ":PRINT"■ 1"
 813 PRINT"■  ":PRINT"■ 0"
 831 PRINT"■  "
 832 PRINT"■ ■                A"
 833 PRINT"■ ■               ■ ■<■ "
 834 PRINT"■ ■               ■   ■"
 840 T=200:GOSUB600
 842 BACE=33743
 845 POKEBACE,22
 850 POKEBACE-40,24
 855 POKEBACE-80,01
 860 T=1:GOSUB600
 870 POKEBACE,34
 880 POKEBACE-40,22
 890 POKEBACE-80,24
 900 POKEBACE-120,01
 910 BACE=BACE-40
 911 GOSUB600
 912 POKEBACE+40,96+128
 913 POKEBACE,34
 914 POKEBACE-40,22
 915 POKEBACE-80,24
 916 POKEBACE-120,01
 917 BACE=BACE-40
 918 IFR=<0THEN995
 920 FORX=1TOR*2
 925 POKEBACE+40,96
 930 POKEBACE,34
 940 POKEBACE-40,22
 950 POKEBACE-80,24
 960 POKEBACE-120,01
 962 REM
 970 BACE=BACE-40
 980 GOSUB600
 990 NEXTX
 995 GOSUB2000:R=0:T=550:GOSUB600
 999 PRINT"■":GOTO1045
 1000 PRINT"■      COMPUTER CHECK LIST ":GOTO1010
 1002 GETA$:IFA$="D"THENRETURN
 1003 GOTO1002
 1010 PRINT"HIT (D) WHEN DONE"
 1020 PRINT"■■■■MAKE SURE INTERFACE IS HOOKED TO THE   --COMPUTER":GOSUB1003
 1030 POKE59471,1:PRINT"■REWIND THE TAPE ALL THE WAY":GOSUB1002
 1035 PRINT"■PRESS PLAY ON TAPE":POKE59471,0:GOSUB1002
 1040 PRINT"■I'M NOW MOVING TO BEGINNING OF THE RECO--RDING":G(0)=1000:GOSUB480
 1045 POKE59471,0:GOSUB640:GOSUB2222
 1050 PRINT"■SECTION #",S(2),,"GROUP#",G(1):INPUT"DO YOU WISH TO GO ON";A$
 1060 IFA$="NO"THEN1080
 1061 GOTO24
 1080 INPUT"DO YOU WISH TO START FROM BEGINNING OF WORDS";A$:IFA$="NO"THEN1083
 1081 RESTORE:GOSUB640
 1082 GOTO24
 1083 INPUT"DO YOU WANT TO GO TO THE NEXT SECTON";A$
 1084 IFA$="NO"THEN1086
 1085 PRINT"I'M MOVING TO BEGINING OF THE SECTION":G(0)=200*W:GOSUB480:GOTO1045
 1086 INPUT"DO YOU WANT TO CHANGE THE PACE";A$:IFA$="NO"THEN1090
 1087 GOSUB2222:GOTO1050
 1090 INPUT"DO YOU WANT TO LOOK THRU THE SECTON OF WORDS";A$:IFA$="NO"THEN1100
 1095 GOTO725
 1100 INPUT"■DO YOU WANT TO MAKE A TAPE FOR THE COMP--UTER";A$:IFA$="YES"THEN500
 1110 INPUT"■DO YOU WANT TO VERIFY (TAPE-WORD)■SYNC.";A$
 1130 PRINT"■■■■■■■■■■BYE"
 1140 END
 2000 REM DRAW LINE****
 2001 K=-36  :IFR=0THEN2060
 2003 HEAD=BACE-93
 2010 FORX=0TO12
 2020 POKEHEAD+X,45
 2030 T=3:GOSUB600
 2040 NEXTX
 2050 RETURN
 2060 HEAD=33570:GOTO2010
 2222 REM CHANGE PACE WITHOUT ALTERING   DATA STATEMENTS---
 2225 PRINT:PRINTTAB(15);"?PACE CHANGE?"
 2226 PRINT"TO PICK A NEW PACE +5 IS THE SLOWEST AND -5 IS THE FASTEST "
 2230 INPUT"TYPE (0) TO PUT IT BACK";VAN
 2235 IFVAN>5ORVAN<-5THEN2226
 2240 PET=VAN*2
 2245 IFPET>10ORPET<-10THEN2225
 2250 RETURN
 2500 RESTORE:REM VERIFY (WORD-TAPE) SIC.---
 2504 PRINT"■GET INS. OUT OF PAMTHLET"
 2515 PRINT"■■■■      HIT (D) WHEN DONE READING INS.":GOSUB1002
 2519 INPUT"HOW MANY SECTIONS OF WORDS";S:FORX=1TOS:GOSUB640:FORI=1TOW
 2530 PRINT"■WORD # ";I,,"SECTION # ";S(2),,"GROUP # "G(1)
 2535 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■";W$(I):T=270:GOSUB600:PRINT"■■5":T=35:GOSUB
K
 2542 PRINT"■■■■4":GOSUB600:PRINT"■■■■■■3":GOSUB600:PRINT"■■■■■■■■2":GOSUB600
 2548 PRINT"■■■■■■■■■■1":GOSUB600:PRINT"■■■■■■■■■■■■■■START":GOSUB440
 2555 NEXTI,X
 2570 PRINT"■END OF WORDS":END
 3000 DATA1,1,10,A,AND,AWAY,BIG,BLUE,CAN,COME,DOWN,FIND,FOR
 3010 DATA6,6,6,6,6,6,6,6,6,6
 3020 DATA1,2,10,FUNNY,GO,HELP,HERE,I,IN,IS,IT,JUMP,LITTLE
 3030 DATA6,6,6,6,6,6,6,6,6,6
 3040 DATA1,3,10,LOOK,MAKE,ME,MY,NOT,ONE,PLAY,RED,RUN,SAID
 3050 DATA6,6,6,6,6,6,6,6,6,6
 3060 DATA1,4,10,SEE,THE,THREE,TO,TWO,UP,WE,WHERE,YELLOW,YOU
 3070 DATA6,6,6,6,6,6,6,6,6,6
 3080 DATA2,1,10,ALL,AM,ARE,AT,ATE,BE,BLACK,BROWN,BUT,CAME
 3090 DATA6,6,6,6,6,6,6,6,6,6
 3100 DATA2,2,10,DID,DO,EAT,FOUR,GET,GOOD,HAVE,HE,INTO,LIKE
 3110 DATA6,6,6,6,6,6,6,6,6,6
 3120 DATA2,3,10,MUST,NEWS,NO,HOW,ON,OUR,OUT,PLEASE,PRETTY,RAN
 3130 DATA6,6,6,6,6,6,6,6,6,6
 3140 DATA2,4,10,RIDE,SAW,SAY,SHE,SO,SOON,THAT,THERE,THEY,THIS
 3150 DATA6,6,6,6,6,6,6,6,6,6
 3160 DATA2,5,10,TOO,UNDER,WANT,WAS,WELL,WENT,WHAT,WHITE,WHO,WILL
 3170 DATA6,6,6,6,6,6,6,6,6,6
 3180 DATA3,1,10,WITH,YES,AFTER,AGAIN,AN,ANY,AS,ASK,BY,COULD
 3190 DATA6,6,6,6,6,6,6,6,6,6
 3200 DATA3,2,10,EVERY,FLY,FROM,GIVE,GOING,HAD,HAS,HER,HIM,HIS
 3210 DATA6,6,6,6,6,6,6,6,6,6
 3220 DATA3,3,10,HOW,JUST,KNOW,LET,LIVE,MAY,OF,OLD,ONCE,OPEN
 3230 DATA6,6,6,6,6,6,6,6,6,6
 3240 DATA3,4,10,OVER,PUT,ROUND,SOME,STOP,TAKE,THANK,THEM,THEN,THINK
 3250 DATA6,6,6,6,6,6,6,6,6,6
 3260 DATA4,1,10,WALK,WERE,WHEN,ALWAYS,AROUND,BECAUSE,BEEN,BEFORE,BEST,BOTH
 3270 DATA6,6,6,6,6,6,6,6,6,6
 3280 DATA4,2,10,BUY,CALL,COLD,DOES,DON'T,FAST,FIRST,FIVE,FOUND,GAVE
 3290 DATA6,6,6,6,6,6,6,6,6,6
 3300 DATA4,3,10,GOES,GREEN,IT'S,MADE,MANY,OFF,OR,PULL,READ,RIGHT
 3310 DATA6,6,6,6,6,6,6,6,6,6
 3320 DATA4,4,10,SING,SIT,SLEEP,TELL,THEIR,THESE,THOSE,UPON,US,USE
 3330 DATA6,6,6,6,6,6,6,6,6,6
 3340 DATA4,5,10,VERY,WASH,WHICH,WHY,WISH,WORK,WOULD,WRITE,YOUR,ABOUT
 3350 DATA6,6,6,6,6,6,6,6,6,6
 3360 DATA5,1,10,BETTER,BRING,CARRY,CLEAN,CUT,DONE,DRAW,DRINK,EIGHT,FALL
 3370 DATA6,6,6,6,6,6,6,6,6,6
 3380 DATA5,2,10,FAR,FULL,GOT,GROW,HOLD,HOT,HURT,IF,KEEP,FIND
 3390 DATA6,6,6,6,6,6,6,6,6,6
 3400 DATA5,3,10,LAUGH,LIGHT,LONG,MUNCH,MYSELF,NEVER,ONLY,OWN,PICK,SEVEN
 3410 DATA6,6,6,6,6,6,6,6,6,6
 3420 DATA5,4,10,SHALL,SHOW,SIX,SMALL,START,TEN,TODAY,TOGETHER,TRY,WARM
 3430 DATA6,6,6,6,6,6,6,6,6,6
 3440 END
READY.

READY.
```

Acknowledgement: We are indebted to the Salem Oregon School District for the loan of their PET and for providing the complete Dolch word list, and to CPU of Salem for designing and building the interface as a substitute for the PET accessory we couldn't afford.

From the Makers of THE Basic Switch™ for Old PET® Owners comes

The Spacemaker™
for New PET Owners

No Room For Your ROM?

If you're an owner of a "new-style" PET, you've probably discovered by now that your Commodore Word Processor ROM and your BASIC Programmers Toolkit ROM both go into the same empty socket in your PET. With the Spacemaker,™ you'll simply install both ROMS on the Spacemaker, plug it into your empty ROM socket, and flip a convenient external switch to select either the Toolkit or the Word Pro ROM.

Spacemaker $27.00
ROMdriver $37.00

Switch ROMs Manually or from Software

The ROMdriver™ is a companion device that can drive up to three Spacemakers ... allowing ROM selection on each Spacemaker under software control. Spacemakers may be "daisy-chained" as well, allowing selection from ROMdriver of a single ROM from several Spacemakers. The Spacemaker is designed for both manual switching and software switching, so ROMdriver is not required for use of the Spacemaker in manual mode. Each ROMdriver is supplied with cable kit for 1 Spacemaker. Additional kits, $4.50 each.

Spacemaker and **ROMdriver** will be available at these and other dealers:

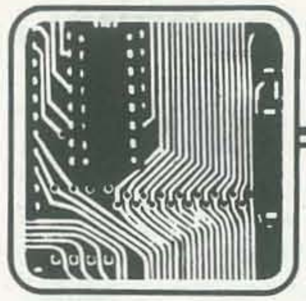
A B Computers
115 E. Stump Road
Montgomeryville, PA 18936

NCE/COMPUMART
270 Third Street
Cambridge, Mass. 02114

New England Electronics Co., Inc.
679 Highland Avenue
Needham, Mass. 02194
or may be ordered directly from

Small System Services, Inc.
900 Spring Garden Street
Greensboro, NC 27403
919-272-4867
M/C Visa Accepted. N.C. Residents add 4% Sales Tax.

Inquire about pricing and availability of Computhink Option, Required for Computhink Disk Drives.



User Inspired Hardware Products

New Directions in Numerical Computing

What happens as more micros get into the hands of the people? We can expect new uses for computers, new styles of software design, and as Harry suggests—new hardware based on the personal computerist's needs. Take a peek into the future with us . . . and tell us what you see! — *RZ*

BY HARRY L. PRUETZ

Now that microprocessors have been in use for several years, heavily numerical software packages are becoming more common. The history of *maxi*computers and minicomputers predicts that specialized microprocessor hardware for numerical applications should start appearing soon. However, the specific design of the hardware might involve completely new concepts because of the sheer number and diversity of personal computer users.

Personal computer users (just three decades after the start of the computer revolution) are about in the same situation as amateur astronomers relative to the centuries-old field of astronomy. Personal computer users do not have the specialized knowledge that many professional people have. But, personal computerists are numerous enough to make important discoveries related to computers, and knowledgeable enough to provide constructive criticism about computer hardware design.

General purpose computers were initially designed for numerical computing. They used a fairly large word size and a narrow definition of integer and floating point number representations. The standard 8-bit byte used in today's microprocessors allows for more possible number representations, using software utility routines. Even "impractical" number representations proposed in the past are perhaps practical today using microprocessors. The most obvious reason for this is that higher level languages (BASIC, APL, PASCAL, FORTRAN, and COBOL) are mostly interpreted languages on microprocessors and are slow when compared with machine coded programs. This fact makes the extra time spent in more involved numerical utility routines relatively small when compared with total computation time.

Another reason is that the amount of time spent on numerical software routines is already high even for the standard integer and floating point representations as used in the past. The longer computation time needed for "impractical" representations would not be excessive. Hopefully, personal computer users now writing programs requiring involved numerical calculations will discover alternative software methods that can be implemented using hardware. Perhaps they will find methods that provide better accuracy at about the same price as the old standard implementations. Proving that a method works in all cases and determining how to implement the method with hardware can be left to the professionals.

Continued from pg. 41

man in his work, a machine which can be totally controlled by each and every one of us. That understanding is what we are trying to achieve.

The issue of computer awareness is the topic of a one-day workshop conducted by the authors. Typical workshop attendees include educators, owners of small businesses, librarians, social workers, and medical professionals.

Unlike traditional computer workshops that stress hands-on encounters with computers, our workshop deals with underlying issues in computer demystification. First we have each participant complete the sentence *"A computer is . . . "*. Next members of the group express their opinions on computers – good or bad. During the process of sharing views with the rest of the group, each member finds that there is a lot of common ground in various people's experiences. The rest of the morning is devoted to understanding what a computer does, where computer jargon originated, and how to cope with computer myths (e.g., how to handle someone who tells you that "the computer made a mistake"). By lunchtime, the major demystification process has been completed. The afternoon is then devoted to a description of different types of computer services (time-sharing, remote batch, personal computers, etc.) and a description of case histories which are relevant to the audience.

In our next article we will delve into some computer myths in more detail. Until then you might want to catalog your feelings about computers and share this article with some friends whose views of these machines might differ from yours.

6502 User Notes, The PET® Gazette and PET User Notes are now a part of

COMPUTE.
The Journal for Progressive Computing™

Continuing major sections on Business, Industrial and Educational Applications and Resources, Plus The PET® Gazette, The ATARI® Gazette, The APPLE® Gazette and The SBC (Single Board Computer) Gazette. All in each issue!

A Sampling of Our 104 page "Super" Fall Issue:

Tokens in Microsoft BASIC: Harvey Herman. ATARI Computers: The Ultimate Teaching Machines?: John Victor. Carl Moser Presents a Universal 6502 Memory Test. Microcomputers in Nuclear Instrumentation: Joe Byrd. AIM 65 Review: Don Clem. Mastering The Ohio Scientific Challenger 1P, A Learn-By-Doing Approach: Keith Russell and Dave Shultz. CORVUS 11A Disk Drive for APPLE: A Review by Michael Tulloch. Pierre Barrette on Microcomputers in Education. Len Lindsay Reviews Three Word Processors. PET in Transition/ROM Upgrade Map: Jim Butterfield. Trace for the PET: Brett Butler. 32K PET Programs Arrive: Len Lindsay. Using Direct Access Files With the Commodore 2040 Dual Disk Drive: Chuck Stuart, plus Reviews, Resources and Products.

New Features Coming in January include: "Rambling" by Roy O'Brien, "The Tape Exchange" by Gene Beals, and the Single Board 6502 by Eric Reinke.

1980 Bimonthly Subscription (Six Issues)	$ 9.00
"Super" Fall Issue With 1980 Subscription	1.00
	$10.00

Make Check or Money Order Payable to **COMPUTE.**
Post Office Box 5119
Greensboro, North Carolina 27403 USA

COMPUTE., the new 6502 resource magazine for PET, Apple, Atari, Kim, Sym, Aim and OSI Owners.

COMPUTE. The Journal for Progressive Computing is published by Small System Services, Inc. of Greensboro, North Carolina. Robert Lock, Editor/Publisher.

THE FURTHER ADVENTURES OF FORTRAN MAN

In our last episode, as the main body of the Underground Resistance Movement battles the Glitchmaster's forces in far off Capital City, Fortran Man is led into the low-level laboratories of Castle McIntel. Here, with the help of an in-circuit emulator, his missing memories (lost when he was inadvertently exposed to UV light while in PROM) are at last restored, with his friend and companion Billy Basic serving as source file!

All seems restored—but wait! Angus McIntel, chief of the Clan and operator of the emulator, had set a data switch inadvertently backwards, and Billy Basic now occupies F-Man's body, and vice-versa!

Most confusing! But fortunately, a slight re-connection and a brief ZAP of the emulator succeeds in switching them back into their proper program spaces, and none too soon, for realtime is running out!

General Wirewound, although old-fashioned in many ways, is still possessed of a greater tolerance than most, and continues to conduct the full power of the rebel attack. Yet, without parallel support from Linea and her resistance decades, the attack is bound to enter a failure mode!

Linea had diverted her march across the data fields to bring F-Man to the McIntel stronghold, knowing that his powers, once restored, would be of invaluable aid in overcoming the Glitchmaster's troops and freeing Microprocessorland from his noisy reign. But now the battle is on, and Castle McIntel is far from the battlefield . . .

Can they make their connection in time to save the Land of the Little People from utter doom?

Volume III Episode 10
BY LEE SCHNEIDER & TODD VOROS

In the memory space of General Wirewound, outside the walls of Capital City, resides a major question—will his support software arrive in time? But then, alerted by a sudden output from one of the outpost elements, all perform an immediate lookup function just as a gigantic, shadowy form feeds over them . . .

Quickly they propagate towards the nearby data field where the great dragon has come down to unload its data cargo, to be met by Linea herself and her entire army!

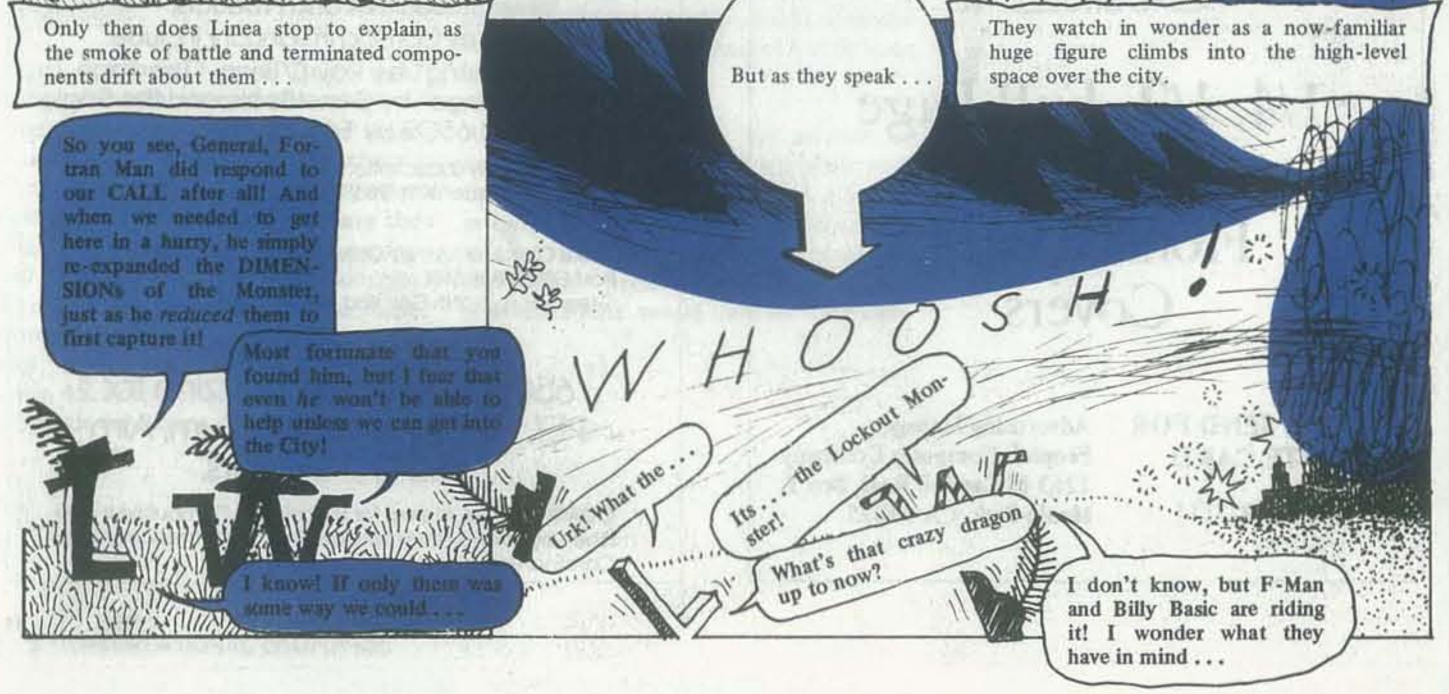
Greetings, Comrade! Your connection is none too soon, for we are already overloaded!
But how . . . ?
No time for that now, General. All nodes, branch off and attack!
Yes . . . and with Linea and her entire Resistance Force . . . *and* Clan McIntel's too!

And attack they do, connecting with the other underground supply lines in their charge against the capacitive guards which hold up the potential on the city walls. They advance ever closer through a deadly hail of high-voltage noise spikes . . .

Only then does Linea stop to explain, as the smoke of battle and terminated components drift about them . . .

But as they speak . . .

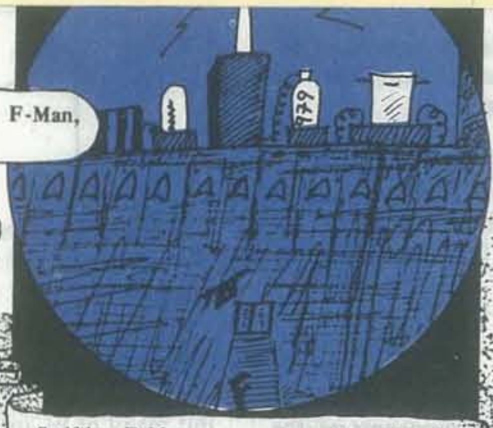
They watch in wonder as a now-familiar huge figure climbs into the high-level space over the city.


So you see, General, Fortran Man did respond to our CALL after all! And when we needed to get here in a hurry, he simply re-expanded the DIMENSIONs of the Monster, just as he *reduced* them to first capture it!
Most fortunate that you found him, but I fear that even *he* won't be able to help unless we can get into the City!
I know! If only there was some way we could . . .
WHOOSH!
Urk! What the . . .
Its . . . the Lockout Monster!
What's that crazy dragon up to now?
I don't know, but F-Man and Billy Basic are riding it! I wonder what they have in mind . . .

he resistance troops joyfully invert selves to gain admittance into the ously impeded city, Linea and the ral rush once again to where the ter has landed.

ant, F-Man! A master am!
No time for performance evaulation now, General. Billy, you remainder here with Linea and assist the General with dividing the city—and I'm going to deal with the Glitchmaster myself!
eep?

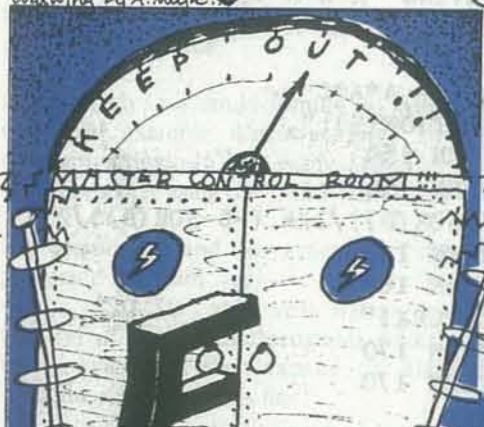
All right Comrade, if anyone can handle the Glitchmaster's file, *you* can! But look out—the Glitch family has a notorious reputation for being sneaky!
In other bytes, F-Man, be careful!
I know, Billy! See you later, folks!
drawing by A. Muja



Swiftly F-Man propagates through the city towards its center—the great, dark, ominous structure from which the Glitchmaster rules his empire, radiating his overpowering noise to all the parts in the land—Tesla Tower!

and unsupported by any other code, Hero branches without hesitation up the staircase generator and into the halls of the Tower itself.

Strange . . . the place seems deserted! The guards must have run off!



Ignoring the hall effect he travels swiftly past rooms filled with noise generators, ground loops, relay coils and other data-destruction devices, until he reaches the very center of the Stronghold—the Master Control room!

Without PAUSE he opens the portal with great force, flinging open the massively shielded doors as he CALLs to his opponent . . .

All right Glitchmaster, your timer has expired! Come on out, and prepare to meet your END!
WHAM!

But then he executes a quick halt—as a calm voice addresses him from across the room . . .

And what is this? Has Fortran Man at last met a super-foe of such power that he dare not attempt to overcome him? Or will there be a battle at all? Will the Land of the Little People ever be clear and noise-free again?

For the answer to this and other non-relocatable questions, tune in again next episode—same cable, same pin number!

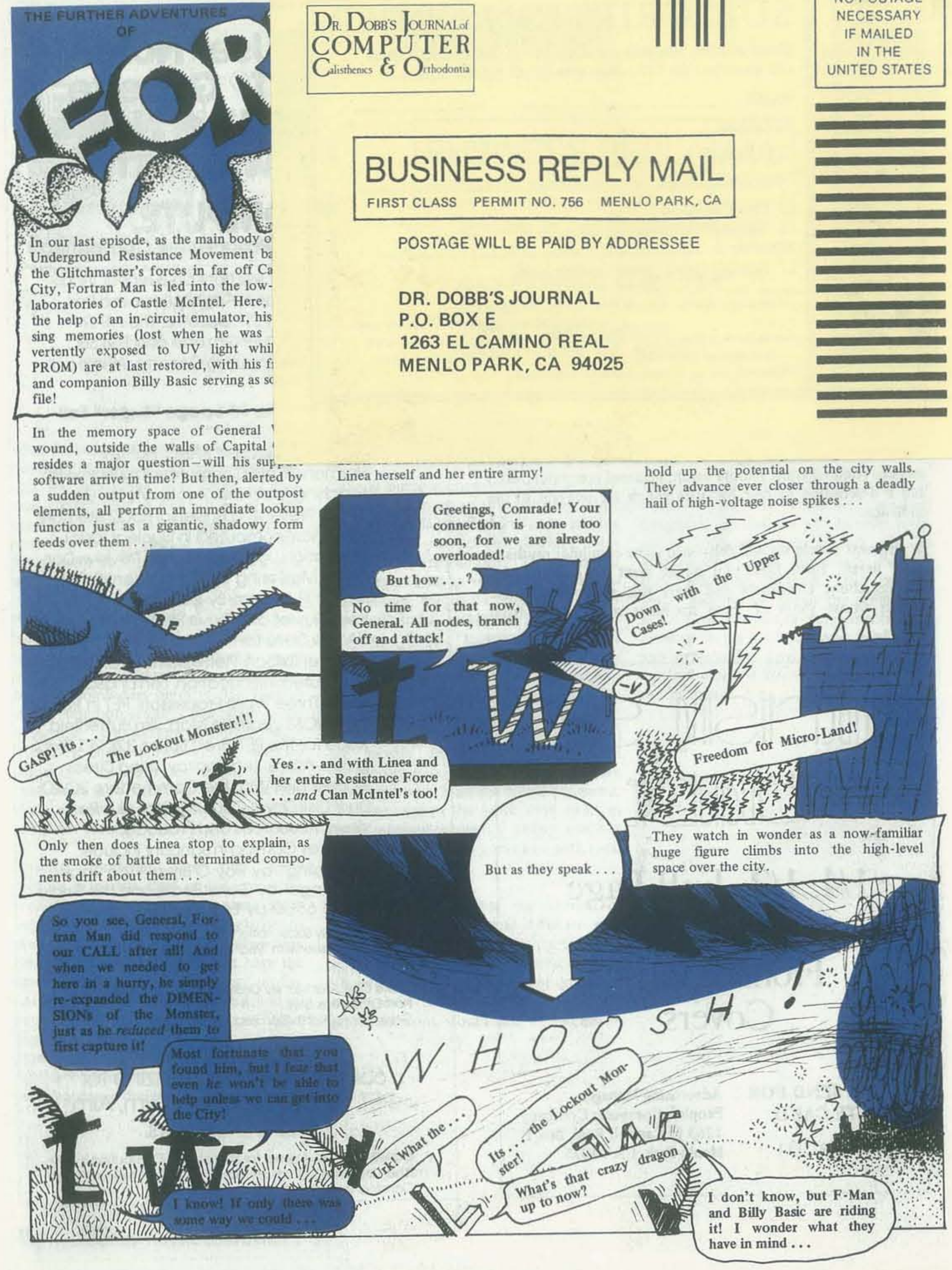
DR. DOBB'S JOURNAL of COMPUTER Calisthenics & Orthodontia

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 756 MENLO PARK, CA

POSTAGE WILL BE PAID BY ADDRESSEE

DR. DOBB'S JOURNAL
P.O. BOX E
1263 EL CAMINO REAL
MENLO PARK, CA 94025


THE FURTHER ADVENTURES OF
FOR
GASP! Its . . .
The Lockout Monster!!!
Yes . . . and with Linea and her entire Resistance Force . . . *and* Clan McIntel's too!
Greetings, Comrade! Your connection is none too soon, for we are already overloaded!
But how . . . ?
No time for that now, General. All nodes, branch off and attack!
Down with the Upper Cases!
Freedom for Micro-Land!
So you see, General, Fortran Man did respond to our CALL after all! And when we needed to get here in a hurry, he simply re-expanded the DIMENSIONs of the Monster, just as he *reduced* them to first capture it!
Most fortunate that you found him, but I fear that even *he* won't be able to help unless we can get into the City!
I know! If only there was some way we could . . .
WHOOSH!
Urk! What the . . .
Its . . . the Lockout Monster!
What's that crazy dragon up to now?
I don't know, but F-Man and Billy Basic are riding it! I wonder what they have in mind . . .

In our last episode, as the main body o[…] Underground Resistance Movement ba[…] the Glitchmaster's forces in far off Ca[…] City, Fortran Man is led into the low-[…] laboratories of Castle McIntel. Here, [...] the help of an in-circuit emulator, his [...] sing memories (lost when he was [...] vertently exposed to UV light whil[...] PROM) are at last restored, with his f[...] and companion Billy Basic serving as s[...] file!

In the memory space of General [...] wound, outside the walls of Capital [...] resides a major question—will his support software arrive in time? But then, alerted by a sudden output from one of the outpost elements, all perform an immediate lookup function just as a gigantic, shadowy form feeds over them . . .

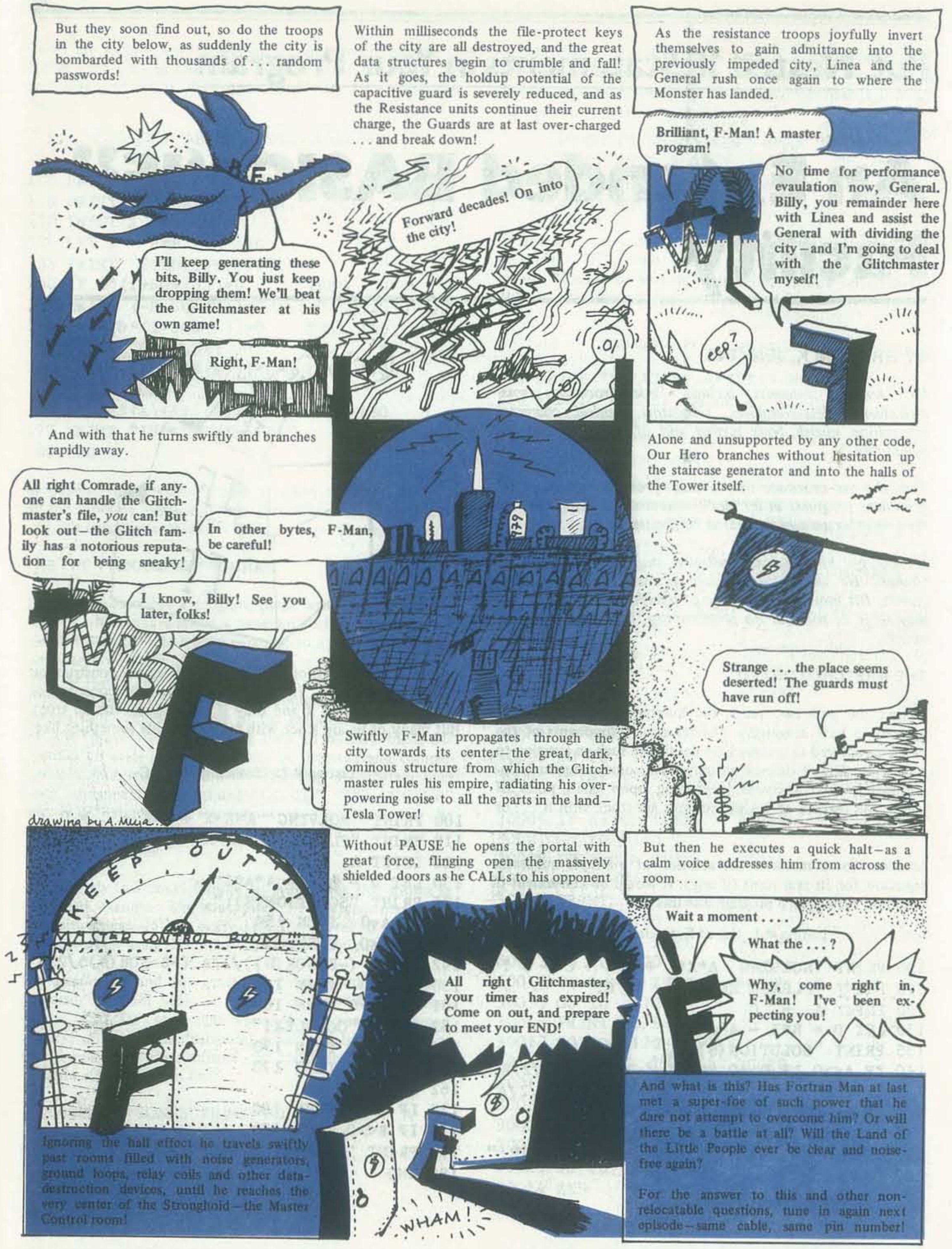
Linea herself and her entire army!

hold up the potential on the city walls. They advance ever closer through a deadly hail of high-voltage noise spikes . . .

Only then does Linea stop to explain, as the smoke of battle and terminated components drift about them . . .

But as they speak . . .

They watch in wonder as a now-familiar huge figure climbs into the high-level space over the city.


I'll keep generating these bits, Billy. You just keep dropping them! We'll beat the Glitchmaster at his own game!
Right, F-Man!
Forward decades! On into the city!
Brilliant, F-Man! A master program!
No time for performance evaulation now, General. Billy, you remainder here with Linea and assist the General with dividing the city—and I'm going to deal with the Glitchmaster myself!
eep?
All right Comrade, if anyone can handle the Glitchmaster's file, *you* can! But look out—the Glitch family has a notorious reputation for being sneaky!
In other bytes, F-Man, be careful!
I know, Billy! See you later, folks!
drawing by A. Muir
Strange . . . the place seems deserted! The guards must have run off!
KEEP OUT
MASTER CONTROL ROOM!!!
All right Glitchmaster, your timer has expired! Come on out, and prepare to meet your END!
WHAM!
Wait a moment . . . .
What the . . . ?
Why, come right in, F-Man! I've been expecting you!

But they soon find out, so do the troops in the city below, as suddenly the city is bombarded with thousands of . . . random passwords!

And with that he turns swiftly and branches rapidly away.

Within milliseconds the file-protect keys of the city are all destroyed, and the great data structures begin to crumble and fall! As it goes, the holdup potential of the capacitive guard is severely reduced, and as the Resistance units continue their current charge, the Guards are at last over-charged . . . and break down!

Swiftly F-Man propagates through the city towards its center—the great, dark, ominous structure from which the Glitchmaster rules his empire, radiating his overpowering noise to all the parts in the land—Tesla Tower!

Without PAUSE he opens the portal with great force, flinging open the massively shielded doors as he CALLs to his opponent

Ignoring the hall effect he travels swiftly past rooms filled with noise generators, ground loops, relay coils and other data-destruction devices, until he reaches the very center of the Stronghold—the Master Control room!

As the resistance troops joyfully invert themselves to gain admittance into the previously impeded city, Linea and the General rush once again to where the Monster has landed.

Alone and unsupported by any other code, Our Hero branches without hesitation up the staircase generator and into the halls of the Tower itself.

But then he executes a quick halt—as a calm voice addresses him from across the room . . .

And what is this? Has Fortran Man at last met a super-foe of such power that he dare not attempt to overcome him? Or will there be a battle at all? Will the Land of the Little People ever be clear and noise-free again?

For the answer to this and other non-relocatable questions, tune in again next episode—same cable, same pin number!

Put Nested IF Statements in Your Program

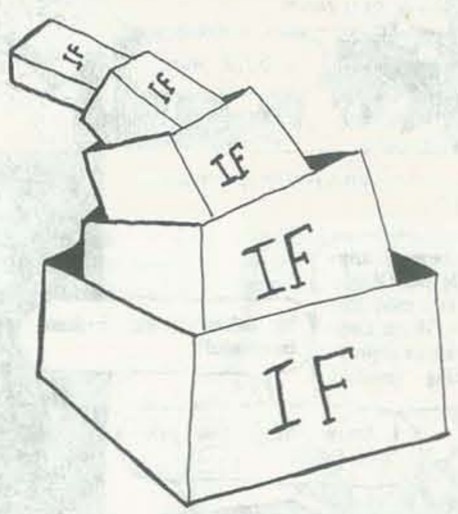
An Extended BASIC "IF" Facility

BY GRAHAM K. JENKINS

Mr. Jenkins' interests include "microprocessor cross-assemblers, cross-compilers, simulators, optical character recognition, squash, body-surfing, and, of course, recreational computing."

With the pre-processor that Graham presents here, you can formulate programs in a "block-structured" BASIC, and let the computer unravel the nested "IF" statements.

If you get by Clayton, Australia, stop and say hello to Graham. He works for Telecom Australia Research Laboratories. Hit him up for a game of squash, but beware! If his playing is as good as his programming, don't wager on the match . . . – *RZ*



THE WRETCHED "GO TO"

During the past two years, the humble GO TO statement appears to have acquired a distinct odor. Proponents of the block-structured languages have developed such an aversion to it that we find it deleted entirely from some compilers! Unfortunately (or otherwise, depending upon one's point of view), the owners of BASIC systems are stuck with it . . . or are they?

Consider for a moment the solution of a simple quadratic equation for its real roots (if any). It would be convenient if we could write such a program like this:

Listing # 1: Use of Extended "IF"

```
100 PRINT "SOLVING  A*X*X + B*X + C = 0 !"
110 PRINT "PLEASE ENTER A, B, C;"
120 INPUT A, B, C
130 LET D = B*B - 4*A*C
135 PRINT "SOLUTION(S):",
140 IF A<>0 IF D>=0 PRINT (-B-SQR(D))/2/A,
                          (-B+SQR(D))/2/A
150 IF A<>0 IF D<0  PRINT "COMPLEX!"
160 IF A=0  IF B<>0 PRINT -C/B
170 IF A=0  IF B=0  PRINT "INDETERMINATE!"
180 END
```

In fact, some versions of BASIC allow this sort of construction (where the repeated "IF" clause means that the consequence should be executed if and only if both conditions are true). But many of us are stuck with having to write something like:

Listing # 2: "Standard" BASIC

```
100 PRINT "SOLVING  A*X*X + B*X + C = 0 !"
110 PRINT "PLEASE ENTER A, B, C;"
120 INPUT A, B, C
130 LET D = B*B - 4*A*C
135 PRINT "SOLUTION(S):",
140 IF A=0  THEN 150
141 IF D<0  THEN 150
142 PRINT (-B+SQR(D))/2/A,(-B-SQR(D))/2/A
150 IF A=0  THEN 160
151 IF D>=0 THEN 160
152 PRINT "COMPLEX!"
160 IF A<>0 THEN 170
161 IF B=0  THEN 170
162 PRINT -C/B
170 IF A<>0 THEN 180
171 IF B<>0 THEN 180
172 PRINT "INDETERMINATE!"
180 END
```

There are shorter ways of writing the program in "standard" BASIC; the tests in lines 150, 160 and 170 for instance, are somewhat redundant. We could alternatively have something like:

Listing # 3: 'ABS" and "SGN" Application

```
100 PRINT "SOLVING  A*X*X + B*X + C = 0 !"
110 PRINT "PLEASE ENTER A, B, C;"
120 INPUT A, B, C
130 LET D = B*B - 4*A*C
135 PRINT "SOLUTION(S):",
140 IF A*(1+SGN(D))=0         THEN 150
142 PRINT (-B+SQR(D))/2/A,(-B-SQR(D))/2/A
150 IF A*D*(1-SGN(D))=0       THEN 160
152 PRINT "COMPLEX!"
160 IF (1-ABS(SGN(A)))*B=0 THEN 170
162 PRINT -C/B
170 IF ABS(A)+ABS(B)>0        THEN 180
172 PRINT "INDETERMINATE!"
180 END
```

In passing, it is worth remarking upon the general utility of the absolute value and sign functions for the testing of multiple conditions, as shown in the example above.

THE PRE-PROCESSOR PROGRAM

This article describes a program (itself written in "standard" BASIC) which will convert a program such as that shown in our first example into a form palatable to a "standard" BASIC compiler/interpreter. Ideally, such a pre-processor would yield an output like that shown in our second example. (The program would be even better if it contained an optimiser to automatically eliminate some of the redundant tests.) There are two problems in producing such an output. First, the operator in each comparison must be detected and complemented. And second, some re-numbering of program lines (with adjustment in associated "GO TO", "IF", "GOSUB" and "ON" statements throughout the program) may be necessary. Neither of these is insurmountable, but they do complicate the issue.

Alternatively, we could require an output like that shown in our third example. Unfortunately, "IF" statements which compare strings do not easily lend themselves to automatic decomposition using the "ABS" and "SGN" functions.

For simplicity, and for generality of application, our pre-processor program replaces each extended "IF" statement with a "GO TO" statement whose destination lies beyond the end of the original program. Additional lines inserted at the program end then expand the statement appropriately. Thus the single program line:

200 IF N$ = "JONES" PRINT "FOUND ! AGE = ";A

might become

200 GOTO 30000

and (if there is room) a remark containing the replaced statement would be inserted immediately thereafter. The following lines would be appended to the program (where 210 is the number of the line found to follow 200).

```
30000 IF N$="JONES" THEN 30002
30001 GOTO 210
30002 PRINT "FOUND! AGE = ";A
30003 GOTO 210
```

If the sequel to an extended "IF" clause is itself another extended "IF" clause, then it also is progressively decomposed. This can be seen in Listing # 4, which is the output of the pre-processor program acting on our first example.

Listing # 4: Pre-processor Output

```
100 PRINT "SOLVING  A*X*X + B*X + C = 0 !"
110 PRINT "PLEASE ENTER A, B, C;"
120 INPUT A, B, C
130 LET D = B*B - 4*A*C
135 PRINT "SOLUTION(S):",
140 GO TO    30000
141 REM IF A<>0 IF D>=0 PRINT
            (-B-SQR(D))/2/A,(-B+SQR(D))/2/A
150 GO TO    30006
151 REM IF A<>0 IF D<0  PRINT "COMPLEX!"
160 GO TO    30012
161 REM IF A=0  IF B<>0 PRINT -C/B
170 GO TO    30018
171 REM IF A=0 IF B=0 PRINT "INDETERMINATE!"
180 STOP
30000 IF A<>0  THEN    30002
30001 GO TO  150
30002 IF D>=0  THEN    30004
30003 GO TO  150
30004 PRINT (-B-SQR(D))/2/A,(-B+SQR(D))/2/A
30005 GO TO  150
30006 IF A<>0  THEN    30008
30007 GO TO  160
30008 IF D<0   THEN    30010
30009 GO TO  160
30010 PRINT "COMPLEX!"
30011 GO TO  160
30012 IF A=0   THEN    30014
30013 GO TO  170
30014 IF B<>0  THEN    30016
30015 GO TO  170
30016 PRINT -C/B
30017 GO TO  170
30018 IF A=0   THEN    30020
30019 GO TO  180
30020 IF B=0   THEN    30022
30021 GO TO  180
30022 PRINT "INDETERMINATE!"
30023 GO TO  180
30024 END
```

STATEMENTS PROCESSED

The types of extended "IF" statement which can be accommodated are shown at the beginning of Listing # 5. This range can be reduced or extended according to requirement merely by adjusting the list of key words appearing in the subsequent "DATA" statements, and effecting a corresponding adjustment to the loop termination value in line 736. The number of "IF" conditions which can appear within an extended "IF" statement is limited only by the user's maximum allowable line width.

INITIALIZATION

During the initialization phase, the invariant data is set up, and the input and output files are attached. This segment of the program may require modification according to the target machine. In our implementation (for an H6000) the "CHANGE" statement is employed to translate an ASCII decimal value for the quotation character into an equivalent string, and the "DELIMIT" statement alters the normal input delimiter character to a carriage return. The "SCRATCH" statement prepares a file for output, and the "MARGIN" statement sets the column after which line wrap-around will occur during output.

It is worth noting that all our output (whether a normal program line, or an extra to be appended at the program end) is directed to the same file, since sorting by line number can be brought about at execution time by an appropriate (pseudo-input) system directive. In those systems where this cannot easily be effected, appended output lines should be directed to a separate file whose entire content can then be read back and included at the end of the output file immediately prior to pre-processor termination.

LINE INPUT

Line input is effected (through the subroutine starting at line 400) using a two-line buffer, so that the number of the next line is available as a "GO TO" destination when required. When an "END" statement is encountered, it is replaced by a "STOP" (as a final exit point) so that branches to its line number are compiled/interpreted corrected. An "END" statement is then appended at the program end.

Each line is checked to determine whether it commences in "IF". If it does, the subroutine beginning at line 700 is invoked to seek a subsequent key word (from the data list). If no key word is discovered, the statement is treated as a normal "IF" statement, and written directly to the output file.

EXTENDED "IF" BREAK-UP

When an extended "IF" statement is found, a "GO TO" and (if room) a remark containing the original contents are inserted in its place as previously discussed. The statement contents up to the first key word are extracted and reconstituted with a "THEN" sequel for insertion at the program end. The original statement sequel is then examined and either included verbatim at the program end, or broken down further as required. Reference to the example shown earlier should clarify the matter. The terminal "GO TO" statement is omitted if the preceding statement sequel was a "STOP" or a "RETURN" (since it would then be redundant).

The astute reader may observe that no syntax checking has been performed upon extended "IF" lines. The reason for this is that the pre-processor only breaks up such lines using the designated key words as delimiters. Any syntax errors will be detected at execution time by the compiler/interpreter. This approach leads to a considerable simplification.

SAVING SPACE

The memory requirement of the pre-processor program itself can be reduced by omission of remarks, and of lines included solely for speed enhancement (e.g. lines 740 through 746). Further reductions can be effected by explicit dimensioning of the L$ and N arrays, or replacing their elements everywhere with equivalent discrete variables. The computation of intermediate variables (necessitated by string-handling limitations of the available compiler) can also be averted in a number of places throughout the program.

The size of the output program can be reduced by elimination of all remarks and spaces (except those within quotation marks). This compression will compromise readability, but the user will (of course!) have retained the original source for reading/modification purposes.

MAKING IT FASTER

Execution speed should be satisfactory on most dedicated systems. Users who pay for processor seconds may, however, wish to enhance the speed. Some available avenues include the elimination of indexed variables (as discussed above), and of redundant computations by the judicious introduction of intermediate variables.

The most critical area for speed enhancement is probably the "next key word" subroutine commencing at line 700. Depending on implementation, it may prove profitable to store the key words in an array instead of repetitively reading them as data. The subroutine does a comparison against the first letter of a key word, then against the whole word. Comparison on a letter by letter basis may be beneficial.

A great saving in execution time can be effected if all non-string characters are eliminated during line input. Key words can be sought without resort to the "string search" subroutine commencing at line 600.

CONCLUSION

The sensible application of the pre-processor program described in this article should lead to vast savings in program development time, and/or to a significant increase in productivity. A number of the modifications discussed can provide you with a range of capabilities appropriate to your particular applications.

Listing # 5: The Pre-Processor Program

```
100 REM   EXTENDED "IF" STATEMENT FACILITY IN BASIC,
102 REM   IMPLEMENTED BY G JENKINS DURING JULY 1979.
104 REM
106 REM   THIS PROGRAM READS BASIC PROGRAMS CONTAINING
108 REM   EXTENDED "IF" STATEMENTS IN THE FOLLOWING
110 REM   FORMS -
112 REM      IF ...   IF     ...
114 REM      IF ...   ON     ...
116 REM      IF ...   LET    ...
118 REM      IF ...   MAT    ...
120 REM      IF ...   READ   ...
122 REM      IF ...   STOP
124 REM      IF ...   INPUT ...
126 REM      IF ...   GOSUB ...
128 REM      IF ...   PRINT ...
130 REM      IF ...   RETURN
132 REM
134 REM   AN EQUIVALENT PROGRAM, CONTAINING ONLY
136 REM   STANDARD BASIC STATEMENTS IS PRODUCED.
138 REM   THE EXTENDED "IF" STATEMENT MAY ITSELF
140 REM   CONTAIN OTHER EXTENDED "IF" STATEMENTS
142 REM   AS SHOWN IN THE FOLLOWING EXAMPLE -
144 REM      IF A>B IF C>D PRINT "A = ";A
146 REM
148 REM   SET UP DATA. Q$ CONTAINS QUOTATION MARK.
150 N(0) = 1
152 N(1) = 34
154 CHANGE N TO Q$
156 DATA "IF", "ON", "LET", "MAT", "READ", "STOP"
158 DATA "INPUT", "GOSUB", "PRINT", "RETURN"
160 REM
162 REM   SOLICIT FILE-NAMES, INITIALLISE FILES -
164 PRINT "EXTENDED ";Q$;"IF";Q$;
166 PRINT " STATEMENT PRE-PROCESSOR;"
168 FILES *;*
170 PRINT "INPUT FILE IS";
172 INPUT L$(1)
174 FILE #1, L$(1)
176 PRINT "OUTPUT FILE IS";
178 INPUT L$(2)
180 DELIMIT #1,(CR)
182 FILE #2, L$(2)
184 SCRATCH #2
186 MARGIN #2, 160
188 REM
190 REM   ADDITIONAL STATEMENTS ARE INSERTED AS
192 REM   NECESSARY, WITH LINE NUMBERS STARTING AT
194 REM   30000. PRE-SET THEIR COUNTER AND GET THE
196 REM   FIRST LINE -
198 N(3) = 30000
200 GOSUB 400
202 REM
204 REM   TRANSFER THE NEXT LINE (IN L$(2), WITH ITS
206 REM   NUMBER IN N(2)) TO THE CURRENT LINE BUFFER,
208 REM   CLEAR QUOTATION FLAG AND CHECK FOR END -
210 L$(1) = L$(2)
212 N(1)  = N(2)
214 Q = 0
216 REM
218 REM   PRE-DEFINE PARAMETERS USED IN STRING
220 REM   SEARCH SUBROUTINE; S1 INDICATES THE START
222 REM   POSITION FOR THE TRANSPARENT SEARCH, AND S2
224 REM   INDICATES THE NUMBER OF CHARACTERS SOUGHT -
226 S1    = 1
228 S2    = 2
230 GOSUB 600
232 IF S$ <> "EN" THEN 244
234 T$ = "STOP"
236 GOSUB 500
238 PRINT #2, N(1); T$
240 PRINT #2, N(3);"END"
242 STOP
244 REM
246 REM   COLLECT THE NEXT LINE, THEN DETERMINE IF
248 REM   THE CURRENT LINE BUFFER COMMENCES IN "IF" -
250 GOSUB 400
252 IF S$ <> "IF" THEN 370
254 REM
256 REM   AN "IF" STATEMENT HAS BEEN FOUND -
258 REM   DETERMINE WHETHER OR NOT IT IS AN
260 REM   EXTENDED ONE BY SEEKING THE NEXT KEY WORD -
262 S1 = S3 + 1
264 GOSUB 700
266 IF S$ = " " THEN 370
268 REM
270 REM   THE CURRENT LINE IS AN EXTENDED "IF"
272 REM   STATEMENT; REPLACE IT WITH A "GO TO"
274 REM   AND (IF ROOM) INCLUDE ITS CONTENTS
276 REM   THEREAFTER AS A REMARK; INDENT THESE
278 REM   NEW STATEMENTS TO SAME LEVEL AS THE
280 REM   ORIGINAL STATEMENT.
282 T$ = "GO TO "
284 GOSUB 500
286 PRINT #2, N(1); T$; N(3)
288 IF N(1)+1 = N(2) THEN 296
290 T$ = "REM "
292 GOSUB 500
294 PRINT #2, N(1)+1; T$; L$(1)
296 REM
298 REM   WRITE 1ST 2 LINES OF DECOMPOSED
300 REM   STATEMENT -
302 S4 = 1
304 T$ = SST( L$(1),S4,S0-S4 )
306 PRINT #2, N(3); T$; " THEN "; N(3)+2
308 PRINT #2, N(3)+1; "GO TO"; N(2)
310 N(3) = N(3) + 2
312 REM
316 REM   IF STATEMENT SEQUEL IS ALSO AN
318 REM   EXTENDED "IF", LOOP TO DO IT AGAIN -
320 IF S$ <> "IF" THEN 336
322 S4 = S0
324 S1 = S3 + 1
326 GOSUB 700
328 IF S$ <> " " THEN 304
330 S0 = S4
332 REM
334 REM   OTHERWISE WRITE SEQUEL AS IS.
336 J = LEN(L$(1)) - S0 + 1
338 T$ = SST( L$(1),S0,J )
340 PRINT #2, N(3); T$
342 N(3) = N(3) + 1
344 REM
346 REM   IF KEYWORD WAS "STOP" OR "RETURN",
348 REM   LOOP FOR NEXT LINE; OTHERWISE WRITE
350 REM   TERMINATING "GO TO" -
352 IF S$ = "STOP"   THEN 202
354 IF S$ = "RETURN" THEN 202
356 PRINT #2, N(3); "GO TO"; N(2)
358 N(3) = N(3) + 1
360 REM
362 REM   END OF EXTENDED "IF" STATEMENT;
364 REM   LOOP FOR NEXT LINE -
366 GO TO 202
368 REM
370 REM
372 REM   THE CURRENT LINE IS STANDARD BASIC -
374 REM   OUTPUT IT, THEN LOOP FOR THE NEXT LINE -
376 PRINT #2, N(1); L$(1)
378 GO TO 202
380 REM
382 REM   *** END OF PROGRAM - SUBROUTINES FOLLOW ***
384 REM   *******************************************
386 REM
388 REM
400 REM   *** LINE INPUT SUBROUTINE ***
402 REM   INPUTS THE NEXT LINE FROM FILE #1, RETURNS
404 REM   LINE NUMBER IN N(2), LINE CONTENTS
406 REM   (EXCLUDING NUMBER) IN L$(2).
408 REM
410 INPUT #1, L$(2)
```

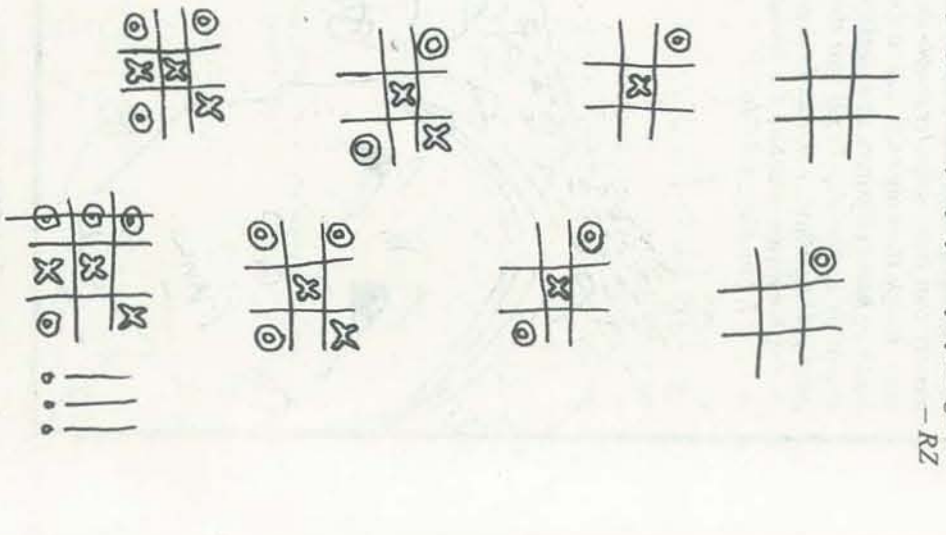
```
412 FOR J = 1 TO LEN(L$(2))
414    T$ = SST( L$(2),J,1 )
416    IF T$ < "0" THEN 422
418    IF T$ > "9" THEN 422
420 NEXT J
422 U$ =  SST(L$(2),1,J-1)
424 N(2) = VAL( U$ )
426 IF T$ <> " " THEN 430
428 J = J + 1
430 L$(2) = SST( L$(2),J,LEN(L$(2))-J+1 )
432 RETURN
434 REM     ********************************************
436 REM
500 REM     *** INDENTATION SUBROUTINE ****************
502 REM     INDENTS CONTENTS OF T$ (BY PREPENDING
504 REM     SPACES) TO SAME LEVEL AS L$(1).
506 FOR J = 1 TO LEN(L$(1))
508    U$ = SST( L$(1),J,1 )
510    IF U$ <> " " THEN 516
512    T$ = U$ & T$
514 NEXT J
516 RETURN
518 REM     ********************************************
520 REM
600 REM     *** STRING SEARCH SUBROUTINE **************
602 REM     RETURNS IN S$ THE NEXT S2 CHARACTERS OF
604 REM     L$(1), STARTING FROM POSITION S1.
606 REM     POSITION IN L$(1) OF LAST CHARACTER
608 REM     FOUND IS RETURNED IN S3.
610 REM     SPACES ARE IGNORED. SEARCH FAILURE RETURNS
612 REM     S$ = " " AND S3 INDETERMINATE.
614 REM
616 S$ = ""
618 FOR S3 = S1 TO LEN( L$(1) )
620    T$ = SST( L$(1),S3,1 )
622    IF T$ = " " THEN 628
624    S$ = S$ & T$
626    IF LEN( S$ ) = S2 THEN 632
628 NEXT S3
630 S$ = " "
632 RETURN
634 REM     ********************************************
636 REM
700 REM     *** NEXT KEY WORD SUBROUTINE **************
702 REM     RETURNS IN S$ THE NEXT KEY WORD IN
704 REM     L$(1), STARTING AT POSITION S1, AND
706 REM     SKIPPING STRINGS DELIMITED BY QUOTATION
708 REM     MARKS (USING FLAG Q). ON EXIT, S0 AND
710 REM     S3 REPECTIVELY CONTAIN START AND
712 REM     FINISH POSITIONS OF S$. SEARCH
714 REM     FAILURE RETURNS S$ = " ".
716 REM
718 IF S1 > LEN(L$(1))-1 THEN 760
720 FOR S0 = S1 TO LEN(L$(1))-1
722    V$ = SST( L$(1),S0,1 )
724    IF V$ = " " THEN 758
726    IF V$ <> Q$ THEN 732
728    Q = 1 - Q
730    GO TO 758
732    IF Q = 1 THEN 758
734    RESTORE
736    FOR T2 = 1 TO 10
738      READ U$
740      REM   FOLLOWING 2 STATEMENTS ENHANCE
742      REM   SPEED - MAY BE OMITTED -
744      S$ = SST(U$,1,1)
746      IF S$ <> V$ THEN 756
748      S1 = S0
750      S2 = LEN(U$)
752      GOSUB 600
754      IF S$ = U$ THEN 762
756    NEXT T2
758 NEXT S0
760 S$ = " "
762 RETURN
764 END
```

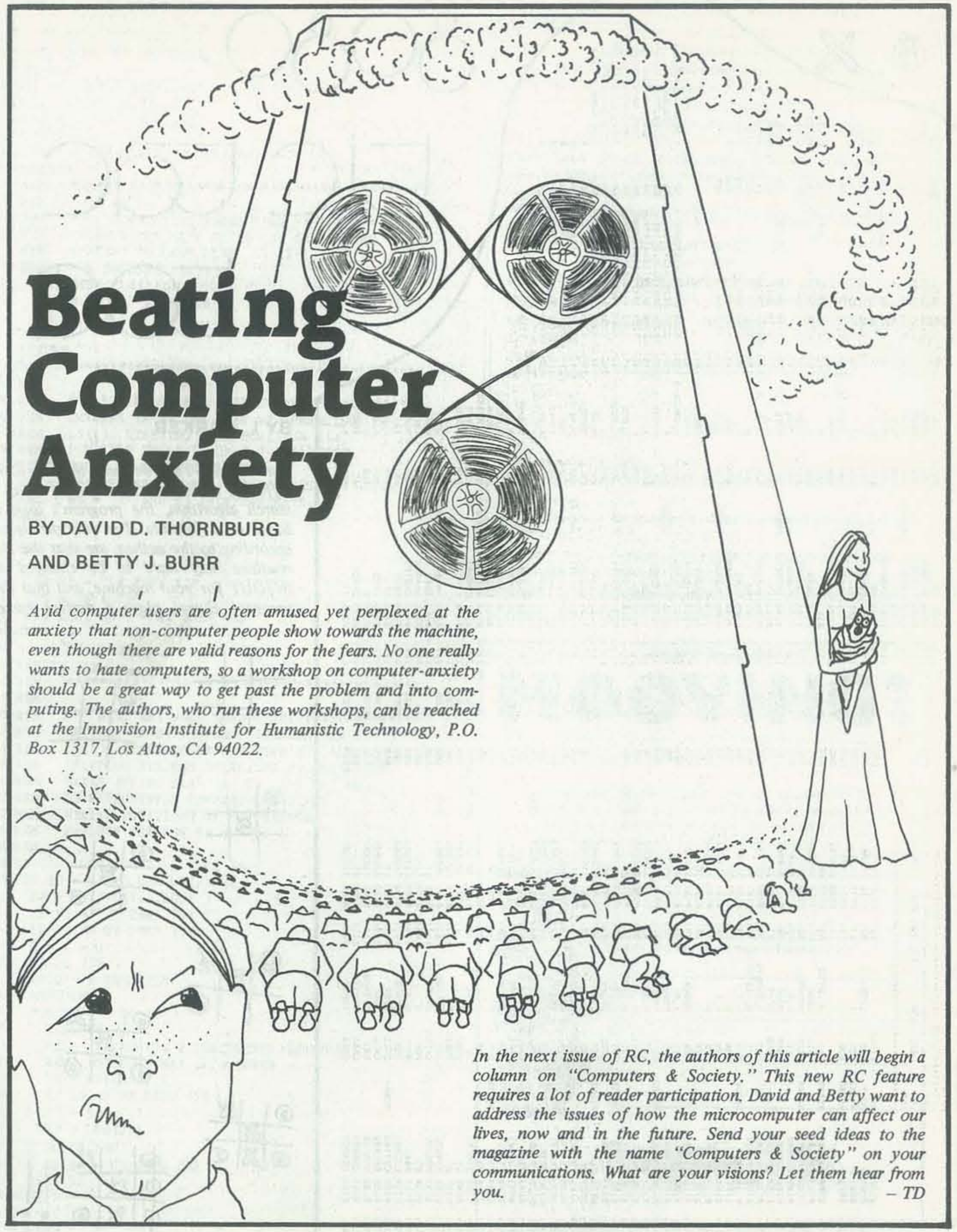
8080: tic tac toe

BY L. BARKER

Here is a Tic-Tac-Toe game that is a little different from most. Instead of using a search algorithm, the program's logic is based on a formula. The only problems, according to the author, are that the I/O routines may have to be changed to IN/OUT for your machine, and that the program always plays a perfect game.
—RZ

```


                ;*************************************
                ;**                                 **
                ;**     8080 CP/M      TIC TAC TOE  **
                ;**                                 **
                ;*************************************

0100            ORG     0100H
FC00 =          PORT    EQU     64512
0001 =          MASKI   EQU     1
0002 =          MASKO   EQU     2

0100 21AA02     START:  LXI     H,INTRO
0103 CD7402             CALL    CRLF
0106 CD6B02     BEGIN:  CALL    OUTPT
0109 210000             LXI     H,0
010C 22EA02             SHLD    TICC
010F 22EC02             SHLD    MOVE
0112 219A02             LXI     H,TTTB+8
0115 010930             LXI     B,256*48+9
0118 79         SLOOP:  MOV     A,C
0119 B0                 ORA     B
011A 77                 MOV     M,A
011B 2B                 DCX     H
011C 0D                 DCR     C
011D C21801             JNZ     SLOOP
0120 3E58               MVI     A,88
0122 329602             STA     TTTB+4
0125 CD7402     BOARD:  CALL    CRLF
0128 0603               MVI     B,3
012A C5                 PUSH    B
012B 219202             LXI     H,TTTB
012E 0E20       SPACE:  MVI     C,32
0130 CD7B02             CALL    PRINT
0133 4E         PLOOP:  MOV     C,M
0134 23                 INX     H
0135 CD7B02             CALL    PRINT
0138 05                 DCR     B
0139 CA4901             JZ      NEXTR
013C 0E20               MVI     C,32
013E CD7B02             CALL    PRINT
0141 0E21               MVI     C,33
0143 CD7B02             CALL    PRINT
0146 C32E01             JMP     SPACE
0149 C1         NEXTR:  POP     B
014A 05                 DCR     B
014B CA5C01             JZ      DONE
014E C5                 PUSH    B
014F E5                 PUSH    H
0150 219B02             LXI     H,LINE
0153 CD6B02             CALL    OUTPT
0156 E1                 POP     H
0157 0603               MVI     B,3
0159 C32E01             JMP     SPACE
015C CD7402     DONE:   CALL    CRLF
015F CD7402             CALL    CRLF
0162 21B702             LXI     H,YOURM
0165 CD6B02             CALL    OUTPT
0168 CD4D02             CALL    INPUT
016B FE31               CPI     49
016D DA5F01             JC      DONE+3
0170 DE31               SBI     49
0172 FE09               CPI     9
0174 D25F01             JNC     DONE+3
0177 4F                 MOV     C,A
0178 219202             LXI     H,TTTB
017B 09                 DAD     B
017C 364F               MVI     M,79
017E 218902             LXI     H,ORDER
0181 09                 DAD     B
0182 CD7402             CALL    CRLF
0185 4E                 MOV     C,M
0186 3AEB02             LDA     TICL
0189 B7                 ORA     A
018A C24402             JNZ     LOSE
018D 3AED02             LDA     TICY
0190 3C                 INR     A
0191 32ED02             STA     TICY
0194 3D                 DCR     A
0195 C29F01             JNZ     SKIPX
0198 79                 MOV     A,C
0199 32EC02             STA     MOVE
019C C3B301             JMP     ONE
019F 3D         SKIPX:  DCR     A
01A0 CAC701             JZ      TWO
01A3 3D                 DCR     A
01A4 CAE601             JZ      THREE
01A7 3AEA02             LDA     TICC
01AA 81                 ADD     C
01AB DE0A               SBI     10
01AD CA3E02             JZ      TIE
01B0 C34402             JMP     LOSE
01B3 3AEC02     ONE:    LDA     MOVE
01B6 07                 RLC
01B7 DE0A               SBI     10
01B9 F2BE01             JP      SKIPR
01BC 2F                 CMA
01BD 3C                 INR     A
01BE 32EA02     SKIPR:  STA     TICC
01C1 32EE02             STA     TICD
01C4 C30D02             JMP     TEST
01C7 3AEE02     TWO:    LDA     TICD
01CA 81                 ADD     C
01CB DE0A               SBI     10
01CD C24402             JNZ     LOSE
01D0 3AEC02             LDA     MOVE
01D3 1F                 RAR
01D4 D20002             JNC     EVEN
01D7 3AEE02             LDA     TICD
01DA 0F                 RRC
01DB 32EA02             STA     TICC
01DE 3E01               MVI     A,1
01E0 32EB02             STA     TICL
01E3 C30D02             JMP     TEST
01E6 3AEE02     THREE:  LDA     TICD
01E9 47                 MOV     B,A
01EA 3AEC02             LDA     MOVE
01ED 90                 SUB     B
01EE C605               ADI     5
01F0 91                 SUB     C
01F1 C24402             JNZ     LOSE
01F4 3AEC02             LDA     MOVE
01F7 80                 ADD     B
01F8 DE05               SBI     5
01FA 32EA02             STA     TICC
01FD C30D02             JMP     TEST
0200 3AEC02     EVEN:   LDA     MOVE
0203 47                 MOV     B,A
0204 3AEE02             LDA     TICD
0207 90                 SUB     B
0208 C605               ADI     5
020A 32EA02             STA     TICC
020D 3AEA02     TEST:   LDA     TICC
0210 B7                 ORA     A
0211 CA2C02             JZ      IMPOS
0214 010800             LXI     B,8
0217 219102             LXI     H,ORDER+8
021A BE         CLOOP:  CMP     M
021B CA2302             JZ      FOUND
021E 2B                 DCX     H
021F 0D                 DCR     C
0220 C21A02             JNZ     CLOOP
0223 219202     FOUND:  LXI     H,TTTB
0226 09                 DAD     B
0227 3658               MVI     M,88
0229 CD2501             CALL    BOARD
022C 21C202     IMPOS:  LXI     H,MESS1
022F CD6B02             CALL    OUTPT
0232 CD4D02             CALL    INPUT
0235 3E59               MVI     A,89
0237 B9                 CMP     C
0238 CA0001             JZ      START
023B C30000             JMP     0
023E 21DA02     TIE:    LXI     H,MESS3
0241 C30601             JMP     BEGIN
0244 21DF02     LOSE:   LXI     H,MESS4
0247 CD6B02             CALL    OUTPT
024A C30001             JMP     START

024D 3A00FC     INPUT:  LDA     PORT
0250 E601               ANI     MASKI
0252 CA4D02             JZ      INPUT
0255 3A01FC             LDA     PORT+1
0258 E67F               ANI     127
025A 4F                 MOV     C,A
025B 3E7E               MVI     A,126
025D 91                 SUB     C
025E DA4D02             JC      INPUT
0261 79                 MOV     A,C
0262 DE20               SBI     32
0264 DA4D02             JC      INPUT
0267 CD7B02             CALL    PRINT
026A C9                 RET

026B 4E         OUTPT:  MOV     C,M
026C 23                 INX     H
026D CD7B02             CALL    PRINT
0270 F26B02             JP      OUTPT
0273 C9                 RET

0274 0E0D       CRLF:   MVI     C,13
0276 CD7B02             CALL    PRINT
0279 0E0A               MVI     C,10

027B 3A00FC     PRINT:  LDA     PORT
027E E602               ANI     MASKO
0280 CA7B02             JZ      PRINT
0283 79                 MOV     A,C
0284 3201FC             STA     PORT+1
0287 B7                 ORA     A
0288 C9                 RET

                ; Data

0289 0409020305ORDER:   DB      4,9,2,3,5,7,8,1,6
0292 4C2E204241TTTB:    DB      'L. Barker'
029B 0D0A2D2D2DLINE:    DB      13,10,'-----------',13,138
02AA 5449432054INTRO:   DB      'TIC TAC TOE',13,138
02B7 594F555220YOURM:   DB      'Your move?',160
02C2 494D504F53MESS1:   DB      'IMPOSSIBLE',13,10
02CE 504C415920MESS2:   DB      'Play again?',160
02DA 5449450D8AMESS3:   DB      'TIE',13,138
02DF 594F55204CMESS4:   DB      'YOU LOSE A',212
02EA 00         TICC:   DB      0
02EB 00         TICL:   DB      0
02EC 00         MOVE:   DB      0
02ED 00         TICY:   DB      0
02EE 00         TICD:   DB      0

02EF                    END 0100H
```





Beating Computer Anxiety

BY DAVID D. THORNBURG
AND BETTY J. BURR

Avid computer users are often amused yet perplexed at the anxiety that non-computer people show towards the machine, even though there are valid reasons for the fears. No one really wants to hate computers, so a workshop on computer-anxiety should be a great way to get past the problem and into computing. The authors, who run these workshops, can be reached at the Innovision Institute for Humanistic Technology, P.O. Box 1317, Los Altos, CA 94022.

In the next issue of RC, the authors of this article will begin a column on "Computers & Society." This new RC feature requires a lot of reader participation. David and Betty want to address the issues of how the microcomputer can affect our lives, now and in the future. Send your seed ideas to the magazine with the name "Computers & Society" on your communications. What are your visions? Let them hear from you. – *TD*

"COMPUTERS!? Yecch!"

"Why should I be interested in computers – all they are doing is making society impersonal."

"So you work with computers. Did I ever tell you about the time my subscription got messed up by a computer?"

"Computers? Sure they're taking over. In a few years babies are going to have computers implanted in their brains at birth. Just wait and see!"

Every person who publicly admits a connection with computers has probably heard some statements like these; statements which perpetuate popular computer myths. This mythology has kept people from learning what computers really do and what computers really cannot do.

While today's youngsters sit down with ease at the keyboard of a personal computer, it has been estimated that one half of the adult community in the U.S. has a basic distrust and fear of computers. This anxiety has not been diminished by increased computer usage in our society. In this article we will describe some of the origins of computer anxiety. We will also describe the structure of a workshop on computer awareness that is conducted by the authors and that includes a great deal of computer demystification. This workshop has been useful in helping many people come to grips with their feelings towards computers and has given these people a basis from which to develop a rational view of this ubiquitous technology.

The importance of a rational understanding of computers is underscored when one looks at the automobile as an analogy. Cars have been around for a long time, and yet there are still people who have "car anxiety" and don't drive. If these people live in a community without public transportation,car anxiety can be a real nuisance. In the computer-rich society of late twentieth century America, computer anxiety will be both a nuisance and a danger.

When computers were solely owned by universities and large corporations, it was possible to ignore negative myths about computers. At that time, very few of us got to see one up close, and even fewer of us got to use one. As regular readers of this magazine know, all that has changed thanks to the advent of the microprocessor. This device not only ushered in an era of improved cost and performance for hundreds of home appliances, but made the low-cost personal computer a reality. In 1978 more than 150,000 personal computers were purchased in the United States alone. It is estimated that 35% of these machines are located in schools. In 1979 the sales of these computers has continued to accelerate. Within a few years home computers will be as commonplace as color television sets. The possibility of vast communication networks is suggested by this potential market. If the future use of such networks is to be decided by the public for the public good, we can ill afford to have many people holding negative views of computers based upon misinformation. All people must become sufficiently computer literate to understand and to participate in determining the role of computers in our society. Computer anxiety *must* be overcome.

Where does computer anxiety come from? There are many sources. Historically, newspapers have tended to present computers either as berserk machines which foul up checking accounts, or as super-sophisticated tools operated by distinguished scientists in white lab coats. While the computer-run-wild stories are almost always traceable to errors on the part of the programmer or computer operator, the media does not find human errors nearly so newsworthy. Having worked around computer scientists for years, we have no idea where the "white lab coat" image came from. It is probably derived from the special treatment large computer systems receive (special air conditioning, special rooms with limited access, etc.). But while large computer installations are housed in special rooms, the vast majority of computers are out in the real world sitting on desks and kitchen tables. Still, many people are afraid that if they touch the machine or push the wrong button "the computer will blow up." Of course, this fallacy is easy to overcome once someone is sufficiently motivated to approach a computer.

Novels and movies have contributed heavily to the idea that computers can run amok and are anthropomorphic (having human-like feelings and responses). To pick just one example, *The Terminal Man* by Michael Crichton is centered around a psychomotor epileptic who is treated by having a computer implanted in his head. This computer monitors his brain waves for an attack and then sends signals to the brain to stop the seizure. One of his doctors wonders why the public has such a negative response to "mind control", especially since "heart control" in the form of pacemakers is considered such a blessing. The book itself answers the question as our hero's computer whips him into a frenzy and he goes on a spree of murders. The anthropomorphic machine is also commonplace in the movies. In *Demon Seed* a computer impregnates a woman (among a shower of sparks no less). These fictionalized accounts of computer behavior do not help computer users gain a proper perspective on these powerful machines.

And if one-sided newspaper reporting and fiction were not enough, a small minority of computer scientists themselves have visions of the future which serve to further alarm the public. As an example, the following quotation by psychologist and computer scientist Chris Evans appeared in the book *Science Fact* which was published in 1977.

> Man is about to create a new companion for himself on his native planet, a companion who will rival and vie with him not for natural resources, but for intellectual supremacy. Man, the undisputed master of the earth, may before too long have to step gracefully aside and yield the reins of power to beings of his own creation.

This message has made its appearance off and on for the last twenty years. Any intelligent computer expert would point out that the word "intellect" means something completely different in the context of machines than it does in people, but what does someone who is "computer illiterate" think when reading this?

Considering the bad press given to computers, it is a miracle that only 50 percent of the public fears them. Yet there is an urgent need for the majority of people to understand that a computer is just another machine which can be used to assist

Continued on pg. 31

SEE WHAT YOU HEAR & HEAR WHAT YOU SEE

PART III

BY HERB MOORE

As it turns out, some of the BASIC programming in this series of articles on the new ATARI may be a little trickier than first thought. Herb found room for improvement in his last program. Even so, these programming ideas present an achievable challenge to beginning and intermediate users of the ATARI computer. In this last article in this series, Herb blends sound with graphics, so you finally "See What You Hear and Hear What You See." – TD

BACK TRACKS FIRST

Finally, I got my hands on an ATARI computer and tried out some of the programs I'd written for the last issue. Yes, that's right, I've been programming in the dark up until now. When, starting backwards as I often do, I tried to run the final program in last issue's article, it made some interesting sounds, but could hardly be recognized as what I was trying to accomplish.

Two simple corrections will straighten things out though. Working backwards from the bottom of the program, the NEXT W was left out of the FOR-NEXT loop in line 7000. It should be:

```
7000  FOR W = 1 TO Y: NEXT W: GOSUB
      8300
```

Also, in order to have all of the voices turn off at the same time, an END statement is necessary at line 230.

```
230 END
```

Since we'll be working with this program again, I'll list it below with corrections shown. One other change I've made here is to make what was line 10 become line 15, to give room to enter some graphics, which we'll do in this article. In this form, the program worked when I ran it. Now that I actually have an ATARI computer, I'll be able to test out my programs before I pass them on.

I learned from this programming experiment that you can cause the machine to do some interesting things by simply leaving lines out of the program. For example, try taking out line 6300, and see what happens. When I did this, the melody was still recognizable as "Ole MacDonald," but Ole Mac seems to be developing a bit of a stutter. If you leave line 6300 out and also remove lines 7000 and 7100 and change the END statement in line 230 to GOTO 15, you'll get an ongoing sequence of sounds that Ole Mac would probably respond to as "darned tootin."

```
15 NO = 243: N1 = 193: N2 = 162: GOSUB 6000
20 N3 = 60: Y = 100: GOSUB 6400
30 N3 = 60: Y = 100: GOSUB 6400
40 N3 = 60: Y = 100: GOSUB 6400
50 N3 = 81: Y = 100: GOSUB 6400
60 GOSUB 8000
70 N0 = 182: N1 = 144: N2 = 121: GOSUB 6000
80 N3 = 72: Y = 100: GOSUB 6400
90 N3 = 72: Y = 100: GOSUB 6400
100 GOSUB 8000
110 N0 = 243: N1 = 193: N2 = 162: GOSUB 6000
120 N3 = 81: Y = 200: GOSUB 6400
130 N3 = 47: Y = 100: GOSUB 6400
140 N3 = 47: Y = 100: GOSUB 6400
150 GOSUB 8000
160 N0 = 162: N1 = 128: N2 = 108: GOSUB 6000
170 N3 = 54: Y = 100: GOSUB 6400
180 N3 = 53: Y = 100: GOSUB 6400
190 GOSUB 8000
200 N0 = 243: N1 = 193: N2 = 162: GOSUB 6000
210 N3 = 60: Y = 400: GOSUB 6400
220 GOSUB 8000
230 END
6000 SOUND 0,N0,10,8
6100 SOUND 1,N1,10,8
6200 SOUND 2,N2,10,8
6300 RETURN
6400 SOUND 3.N3.10,11
7000 FOR W = 1 TO Y: NEXT W: GOSUB 8300
7100 RETURN
8000 SOUND 0,0,0,0
8100 SOUND 1,0,0,0
8200 SOUND 2,0,0,0
8300 SOUND 3,0,0,0
8400 FOR Z = 1 TO 10: NEXT Z
8500 RETURN
```

Let's go on with the once-promised "seeing what you hear and hearing what you see."

ADDING A STAFF

To begin with I'll introduce a program section which gives us a diagram of a musical staff to which we can then add notes. The following section should draw the five lines of the staff and add bar lines with appropriate spaces to later add the notes of the melody with which we've been working.

```
1 REM "MUSIC STAFF"
2 GR. 4
3 COLOR 1
4 SETCOLOR 4, 4, 2
5 REM "STAFF LINES"
6 FOR A = 5 TO 13 STEP 2
7 PLOT 5, A: DRAWTO 60, A
8 NEXT A
9 REM"BAR LINES"
10 B = 5: GOSUB 10000
11 B = 23: GOSUB 10000
12 B = 36: GOSUB 10000
13 B = 51: GOSUB 10000
14 B = 60: GOSUB 10000
10000 PLOT B, 5: DRAWTO B, 13
10010 RETURN
```

Lines 2 thru 4 get us into a color graphics mode, and lines 5 thru 8 will draw the lines of the music staff. You'll remember that STEP 2 in line 6 is telling the machine to count by 2's this is so the computer will put a space between each staff line that it draws.

Lines 9 thru 14 draw the bar lines. I used this way of doing it rather than FOR-NEXT loop, since the bar lines are not evenly spaced due to the different note durations.

At this point some of you might be wondering . . . "What's a bar line?" Sorry, it's not a line in front of your local tavern. In musical jargon, a bar line is the line that separates one measure or "bar" of music from another. The melody we're working with is in 4/4 time. The top 4 in this fraction means that there are four beats to a measure (or bar) and the bottom 4 means that a quarter note gets a count of one beat. Since a single picture is certainly worth a mega-byte of words I'm including another little diagram which should help you out with this.

PLOTTING THE NOTES

So let's begin to integrate this with the sound part of our program. First of all enter the sound part of the program listed in the beginning of this article.

Now add the following lines to the program:

```
19 PLOT 10, 8
29 PLOT 13, 8
39 PLOT 16, 8
48 COLOR 2
49 PLOT 19, 11
78 COLOR 1
79 PLOT 26, 10
89 PLOT 29,10
118 COLOR 2
119 PLOT 32, 11: PLOT 33, 11
128 COLOR 1
129 PLOT 39, 6
139 PLOT 42, 6
168 COLOR 2
169 PLOT 45,
179 PLOT 48, 7
208 COLOR 1
209 PLOT 54, 8: DRAWTO 57, 8
```

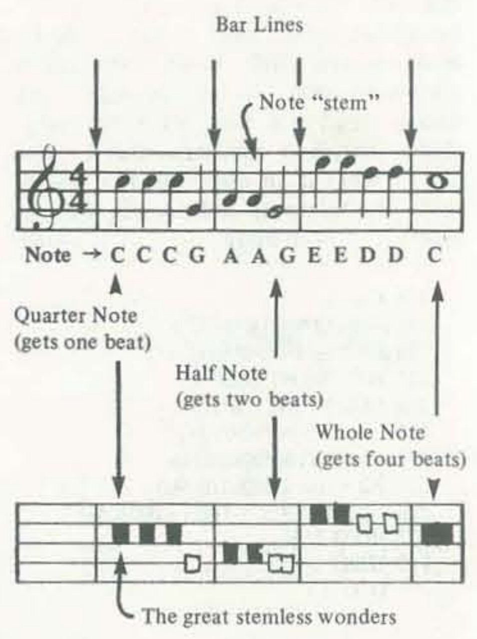
The PLOT statement in each of these new lines we've added plots a point for a given note. What I've done is to put the plot statement for each note in front of the SOUND statment for that note. In this way I am able to take advantage of the time delay in GOSUB 6400 so that each note comes on the screen as the sound is heard.

The COLOR 2 statement in lines 48, 118, and 168 causes the machine to *erase* a point. This is necessary for notes occurring on a staff line. Otherwise, you wouldn't see the note you hear. These notes are shown by blank spaces in the music staff. You then have to use a COLOR 1 statement in lines 78, 128 and 208 in order to plot points in the spaces on the staff.

Rather than using "stems" (see Music Diagram) on these notes I've indicated duration in the following way:

A quarter note = one plotted point (as in line 19)
A half note = two plotted points together (as in line 119)
A whole note = four points together (as in line 209)

Here's a diagram showing "Ole MacDonald" in conventional music notation and in the form this program uses.


Bar Lines
Note "stem"
Note → C C C G A A G E E D D C
Quarter Note (gets one beat)
Half Note (gets two beats)
Whole Note (gets four beats)
The great stemless wonders

A FEW RANDOM THOUGHTS

If you've ever been to a raffle where they put 100 tickets numbered from 1 to 100 in a box and someone reaches in the box to pull out the ticket with the winning number, then you have witnessed the selection of a *random number* between 1 and 100. It's quite easy to get your ATARI computer to pick a random number. You would say something like:

```
10 PRINT RND (1)
```

However, the computer is just as likely to pick a number like 53.85129886 as it is to pick 53. If you want it to skip the "fractional"part (the stuff after the decimal point) you have to tell it to pick a random *integer*. Suppose you wanted to pick, at random one of the 256 notes available on your ATARI computer. You might say:

```
10 NO = INT (256*RND (1) )
6000 SOUND 0, NO, 10.8
```

The part inside of the parentheses says to select a number between 1 and 256 at random. The INT part of this command tells the machine to pick off the integer part of the number (that is, with nothing behind the decimal point). You then, of course, need a SOUND statement as in line 6000 to turn on the sound.

So why all this talk about integers and random numbers? Well, the other day while I was out working in my garden, I was whistling "Ole Macdonald" and this little space ship about the size of a tea saucer landed next to one of the tomato plants and some little green creatures with funny antenna type ears jumped out and began to "sing" along. The best approximation I could get of what it all came out like was to make the following additions to the program we've been using.

```
228 GR. 3
229 FOR Q = 1 TO 100
230 N0 = INT (256*RND (1) )
231 N1 = N0 + 1
232 COLOR INT(16*RND (1) )
233 U = INT(40*RND (1) )
234 V = INT(20*RND(1) )
235 N2 = N0*2:GOSUB 6000
236 N3 = N1*3:Y = 100:GOSUB 6400
237 PLOT U,V
240 NEST Q
250 GOTO 1
```

Line 230 is telling the computer to pick at random, one of the 256 notes available and then play it on voice N0.

Line 231 is making voice N1 play a note at a frequency very close to voice N0. This is what causes the wavy sound you hear in some of the notes.

Line 235 tells the machine to play a note with a value of twice N0.

Line 236 tells it to play a note with a value of three times voice N0.

If you remember from my last article, the lower the note value on the ATARI computer, the higher the frequency of the note. That is, a note value of 10 is actually going to play a higher note than say a value 25 is going to play. So in fact, voice N2 is playing a lower note than voice N0 and voice N3 is playing a lower note yet.

For the graphics part of the program:

Line 228 changes the graphics mode to clear the screen of the staff.
Line 232 tells the machine to pick a random color from the 16 colors available.
Lines 233 and 234 combined with line 237 tell the computer to plot a random point in one of the 20 columns and 40 rows available in graphics mode 3.
Line 250 tells the machine to start the program again at line 1.

SUMMARY

If you're feeling somewhat overwhelmed by all this, don't worry. It will make more sense if you experiment with it, so that you can understand it in your own way. I'm finding that getting the computer to do things is as much a process of experimentation as one of understanding. Or you might say that the understanding comes from experimenting. So keep trying and be sure to let me hear about any ideas for improvements you might have.

TRS 80 SOFTWARE DIRECTORY

THE ESSENTIAL GUIDE TO OVER 5000 TRS 80 PROGRAMS

* ALPHABETIZED & CROSS INDEXED
* INDEXED BY TITLE
* INDEXED BY SUBJECT
* INDEXED BY BASIC
* INDEXED BY VENDORS
* 385 VENDORS NAMES & ADDRESSES
* OVER 120 PAGES
* PUBLISHED SPRING-SUMMER-FALL

SOME OF OUR CUSTOMERS INCLUDE:

ATT	US NAVY	RADIO SHACK	TANDY
IBM	US AIR FORCE	PEPSI COLA	DOCTORS
NBC	UNION CARBIDE	TEXAS INSTRUMENT	INDIVIDUALS

NO CHARGE TO ANY DISTRIBUTORS TO LIST SOFTWARE – SEND US YOUR CATALOG

Single Issue $6.00 Single Foreign Orders Add $2.00 Air
Subscription $15.00 (3 issues) ComputerMat T.M. Subscription Foreign Orders Add $6.00 Air

BOX 1664 DEPT. R LAKE HAVASU AZ 86403 602-855-3357

a New Algorithm for Chess

As the chess game draws closer to end strategy, it naturally becomes more intense. In Part V, Chelberg and Watters continue middle game strategy, then take on piece development, offense and, finally, end strategy. By this time in the series, the articles together are proving to be fine source material for developing your own chess programs. – TD

PART V: MIDDLE GAME STRATEGY – THE STATIC EVALUATION

BY DAVID CHELBERG AND DAVID WATTERS

In this article we will be continuing the description of the middle game strategy. Our last article dealt with the dynamic aspect of the game, and this one will deal with the static evaluation. After the program has evaluated the dynamic position, the cumulative values that have been assigned to each move are examined. If a clear-cut favorite exists, that move is immediately made. If several moves are considered to be better than the rest, only those moves are examined and the remainder are discarded in the interest of saving time. If all values are relatively close, all moves are evaluated statically.

As a general principle, static considerations are of much less importance than dynamic ones. Any individual static consideration is not worth the sacrifice of a pawn or the equivalent material loss. As a cumulative value obtained from this section, the average value assigned to a good positional move is about .8, as compared to a 1 for a pawn capture. Poor positional moves can be reduced as much as 1.5 in value.

The following is a description of the static evaluation sections and the relative weights of each. The section capable of assigning the highest possible value is "attack." This section resembles its dynamic counterpart in that it considers moves that the computer can make to attack its opponent. It is appropriately included in the static evaluation section because the moves are not in response to danger, or direct material gain but rather, are aimed at piece development and at keeping the opponent on the defensive.

In the attack routine we are careful not to reuse any of the attacks which were evaluated as counterattack measures in the dynamic protection routine. If the attacking pieces are assigned their common chess value divided by a fixed amount, our weighting process can be understood. This method is inappropriate for kings though, because it would give a disproportionate weight to king attacks if it were used. This is poor playing for the computer since it would often waste many moves. Instead, attacks on kings are given a flat-rate value that is slightly less than the value given for attacking a queen. It cannot be emphasized enough that random attacks are often more detrimental to the computer's position than helpful. That is why many other factors are considered in the static evaluation.

PIECE DEVELOPMENT

The next most important section is piece development. One of the fundamental rules of middle game strategy involves putting your pieces into controlling positions on the board. This is accomplished by moving them off the back rank. In middle game strategy, our primary concern is the development of bishops and knights: the major pieces should not be used until later in the game. Bishops can be developed by moving them to the center of the board or by *fianchetto* (N2). In either case, a pawn must be moved to permit the development. Good values are assigned to moving those pieces which stand in the way of development. In developing the knight, bishops third is the favored position since it commands most of the center and allows the other pieces to develop. N-R3 places the knight in an isolated position, while N-Q2 or N-K2 blocks the future development of the queen and bishop. All four possible moves are better than staying on the back rank but N-B3 is given the best value. It follows that any piece which is in a developed position ought not to retreat to the back rank.

Many of the considerations involved in piece development are directly related to center control. Our central strategy has two aspects: controlling squares in the center is the greater, while general positioning is the lesser. (See accompanying figure.) In the middle game, the maximum stress rests on the four middle squares, and they should be guarded to the utmost.

Another problem of piece development involves readiness. All pieces ought to be developed properly, not just one side of the board, or a select group of pieces. Every piece should be ready to take its place in the action. One move ought to bring it to the center. So often, in a game of chess, this is not the case. In some cases this is due to the fact that the same pieces are being moved over and over again. Therefore, we assign negative values to moving the same piece successively.

Another cause for slow development is the overmovement of pawns. For this reason, we devote a major section of our static evaluation to pawn structure. In this section of our program, the main goal is to form chains of protected pawns. The ideal chain resembles an inverted "V." Once a chain has been established, there is little need to move those pawns again. Chains should not block those squares attacked by bishops. In order to be able to form stable pawn chains, bad values are given to doubling of pawns as well as breaking chains. And of course, appropriate good values are given to the undoubling of pawns and making chains. In the middle game it is unwise to overextend pawns, so pawn moves beyond the fifth rank are considered poor. Extending a move up two ranks is also poor unless it forms a chain or performs some other useful function. However, in the opening, P-K4 and P-Q4 are good moves to make.

In developing and moving pieces in the middle game, a general rule is to move your lesser-valued pieces more and avoid exposing your queen and higher-valued pieces to attack. This bias is built into our program as knight and bishop moves are prefered to queen and rook moves, which are in turn prefered to pawn moves, which take precedence over king moves. Moving the king should be avoided except when moving to castle.

OFFENSE

The next most vital section is offense. One of the most useful features later in the game is to have your rooks controlling open ranks. Good values are given to moving rooks and queens to open ranks as well as moving pawns off of these ranks to prepare for later control. This preparation is closely regulated by checking for doubling of pawns.

Another important offense is a coordinated attack centered on the opposing king. Pieces that can safely move within one square of the opposing king or that can direct their attack to squares around the enemy king are given a good value. This is an important section since it prevents a random attacking of unguarded pieces. It promotes a coordinated drive for the opposing king. Before this section was implemented, the computer's disoriented attack and lack of direction was very evident. The section provides some semblance of plan for the computer.

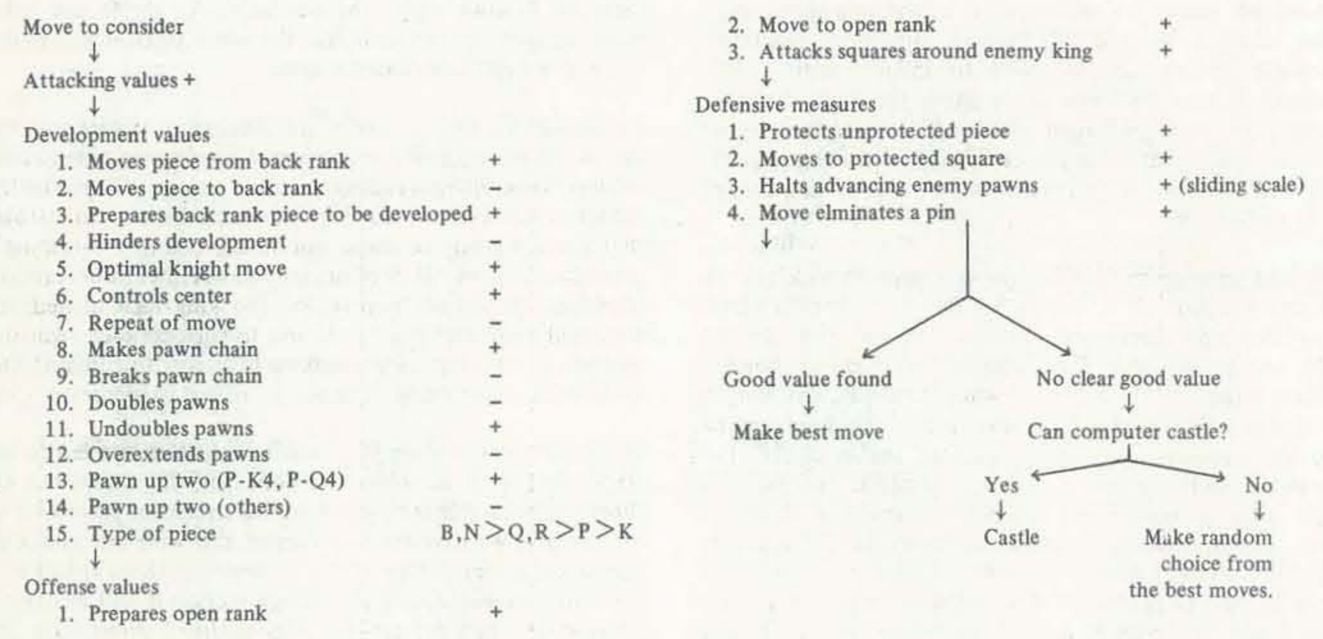
Defensive moves are considered next. These moves include halting enemy pawn advances, eliminating pins and protecting pieces. Halting pawn advances is done on a sliding scale – the closer the pawn to the queening rank the higher the value assigned to stopping it. Eliminating pins is accomplished by looking at a list of pins compiled earlier at the time the "black list" of vital protectors was compiled. If the piece behind the pinned piece can move out, the move is given a good value.

The last defensive goal is to keep as many pieces as possible protected. This enhances the stability of the computer's position. There are two types of moves in this category: those moves that protect an unguarded piece, and those that move an unprotected piece to a guarded square. Both types are given good values.

At this point all moves are compared. If there is no clear decision that can be made with all the factors described to this point now (no cumulative value above 1), the computer will consider castling. It generally tries king's side castling first. If it cannot castle king's side, it attempts queen's side. There is room for improvement in this strategy. We have been considering a castling evaluator to determine if the pawn structure is more favorable for castling on a specific side. This would also take into account the safety of the computer's king in both castled and uncastled positions. If the computer cannot castle, it randomly chooses one among the moves with the highest value.

To summarize the static evaluation, values are assigned on the basis of three major factors: development, offense and defense. Every move is given a value on its developmental features, whether good or bad. Offensive factors, although assigned to only a few moves, probably provide the greatest potential for the computer. Defensive moves provide the least and should only be made if the offensive edge will not be sacrificed. Defensive moves are important, though, in providing a solid foundation for the computer's attack.

Static Evaluation Flow Chart



Move to consider
↓
Attacking values +
↓
Development values

1. Moves piece from back rank	+
2. Moves piece to back rank	–
3. Prepares back rank piece to be developed	+
4. Hinders development	–
5. Optimal knight move	+
6. Controls center	+
7. Repeat of move	–
8. Makes pawn chain	+
9. Breaks pawn chain	–
10. Doubles pawns	–
11. Undoubles pawns	+
12. Overextends pawns	–
13. Pawn up two (P-K4, P-Q4)	+
14. Pawn up two (others)	–
15. Type of piece	B,N $>$ Q,R $>$ P $>$ K

↓
Offense values

1. Prepares open rank	+
2. Move to open rank	+
3. Attacks squares around enemy king	+

↓
Defensive measures

1. Protects unprotected piece	+
2. Moves to protected square	+
3. Halts advancing enemy pawns	+ (sliding scale)
4. Move elminates a pin	+

↓

$$\begin{tikzcd}
 & \cdot \arrow[dl] \arrow[dr] & & \\
\text{Good value found} \arrow[d] & & \text{No clear good value} \arrow[d] & \\
\text{Make best move} & & \text{Can computer castle?} \arrow[dl] \arrow[dr] & \\
 & \text{Yes} \arrow[d] & & \text{No} \arrow[d] \\
 & \text{Castle} & & \text{Make random choice from the best moves.}
\end{tikzcd}$$

Center Control Values

1	0	0	.02	.05	.05	.02	0	0
2	0	0	.02	.05	.05	.02	0	0
3	.03	.03	.05	08	.08	.05	.03	.03
4	.04	.04	.06	.09	.09	.06	.04	.04
5	.04	.04	.06	.09	.09	.06	.04	.04
6	.03	.03	.05	.08	.08	.05	.03	.03
7	0	0	.02	.05	.05	.02	0	0
8	0	0	.02	.05	.05	.02	0	0
	1	2	3	4	5	6	7	8

END STRATEGY

End strategy resembles middle strategy closely, using the same dynamic evaluation routine to find any obvious moves to be made, then continuing in a modified static evaluation. If the correct pieces are on the board, a "formula checkmate" routine may be executed. Otherwise, the static evaluation for middle strategy is expanded to include many other additional factors to help in deciding the move. In end strategy, care must be taken with respect to timing. Moves cannot be made out of sequence without wreaking harmful if not fatal damage. A thorough evaluation is needed to plot the end game course.

In the end strategy evaluation, the emphasis is no longer on piece development. It is hoped by this time that all pieces are satisfactorily developed. Ideally, offense and defense should be in balance. The computer should not send a kamikaze attack on the opponent while leaving its own king in open danger. Neither should it surround its kind with all its pieces like cowboys defending against an Indian attack. The computer should create a balance between offense and defense, altering this balance somewhat in response to who is leading. In our present program, however, the defense outweighs the offense. This is because defensive considerations are much easier to generalize than offensive ones and we have had difficulty determining a set of guidelines for an effective attack. Our end strategy is not as strong as we would like it to be, but we are continuing to improve this area.

In our present version, these are the offensive considerations: eliminate opponent escape squares; advance pawns; and pin the opposing king. Eliminating escape squares is an expansion of the middle strategy section which attacks squares around the king. It restricts the mobility of the opposing king and hopefully prepares for the kill (checkmate). Advancing pawns conflicts with the middle strategy directive of halting advances. However, it is hoped at this time that the computer has sufficiently developed its pieces to support a pawn advance. Piece development is active in this area as are center control and retreat prevention. Placing the opposing king in a pinned or forked position is a means of gaining "free" material. Any material gain possible should be taken advantage of by the leading side. The computer is encouraged to make trades when it's ahead and discouraged from making them when it's behind.

Defensively, the computer has several factors to consider. First, the king should always have an escape square. Pawn structure is closely examined here to guard against exposing the king. If the king is on the back rank, which is usually the case, that rank should be guarded by a rook or queen to prevent the common back-rank-mating trap. To expand the notion of having every piece protected at all times, advancing pawns can be protected by other pawns. In an effort to control the rank, rooks are used to help the pawn.

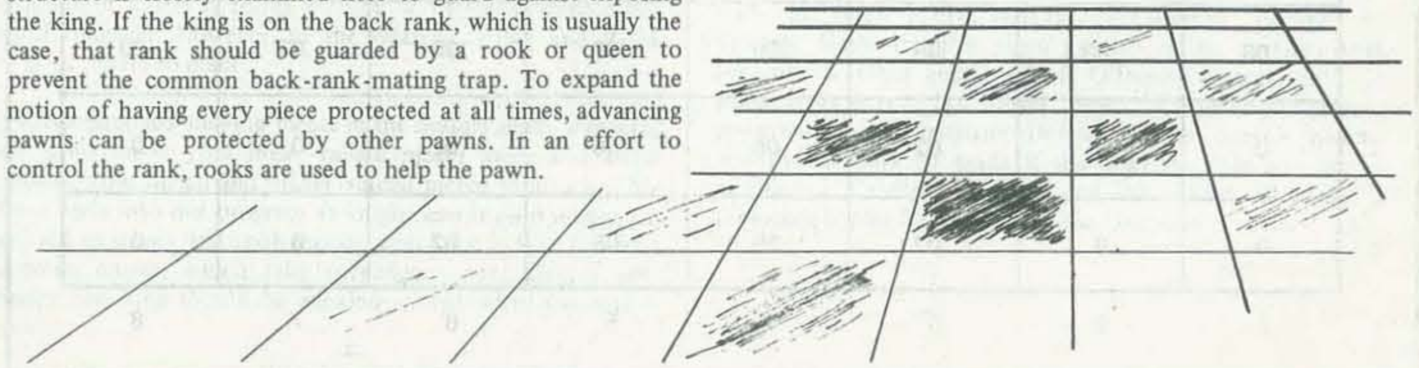
Another factor to consider is guarding against perpetual check. If the computer can put the opponent in check, but cannot accomplish anything constructive by it, then the negative value given to moving the same piece successfully is greatly increased. If the computer is in a hopeless position, a draw by repeated position might be desirable. An ineffective attack is recognized by the fact that the same piece keeps moving without a significant material gain.

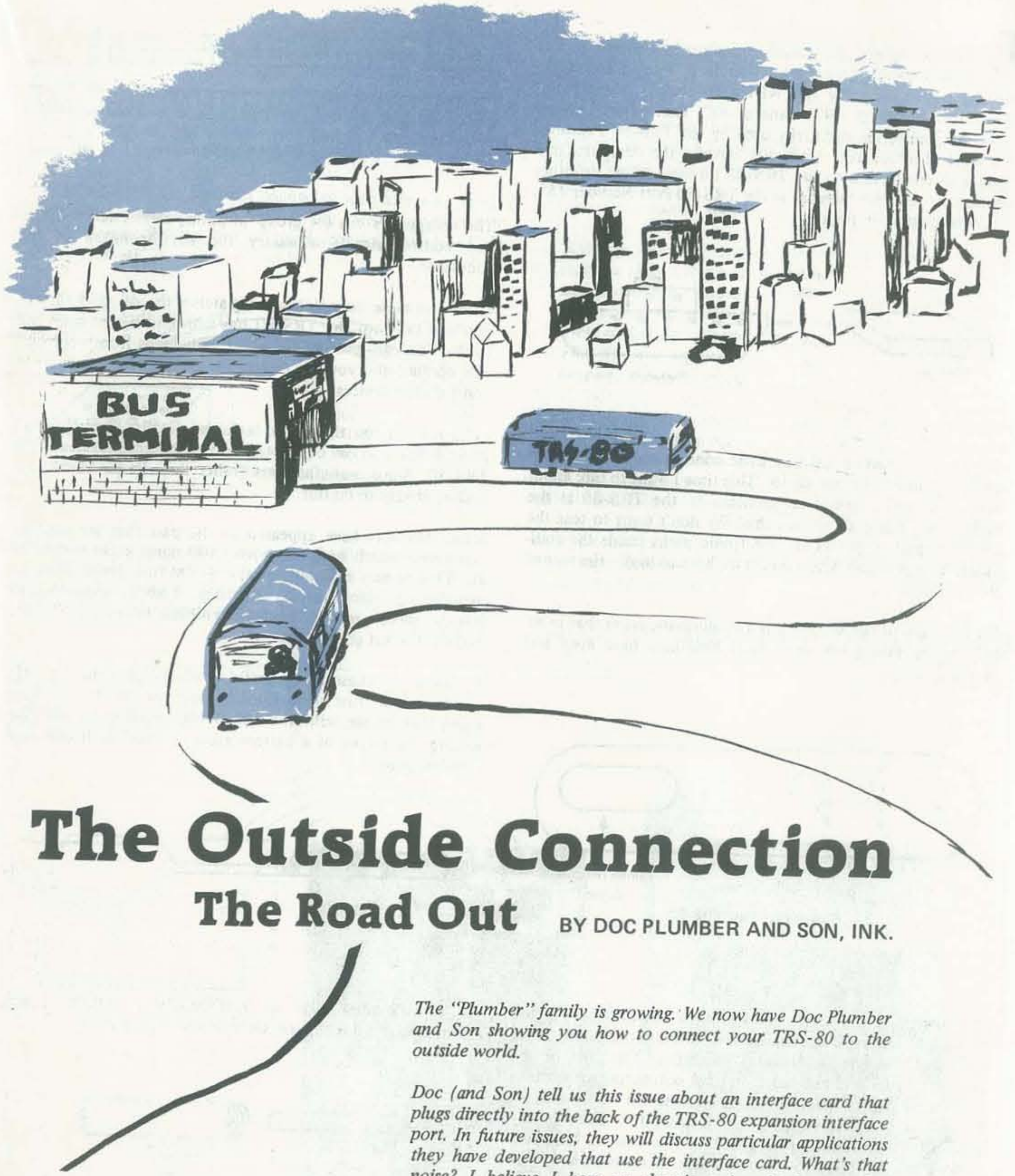
In another section, all moves are checked to ensure that they do not move into stalemate, unless the computer is desperately behind. This prevents ending the game unexpectedly. The final aspect of our end strategy is formula checkmates. In the event that a select group of pieces are on the board, a checkmating procedure is possible. We currently have either in operation or planning king-queen, two rooks, and king-rook checkmates. We could write individual programs to perform each algorithm, and others too, but we are looking for a way to combine them all into a generalized formula in order to conserve space.

In practice, the middle game static evaluation has led to very stable and well designed positions for the computer. One shortcoming in this section is that the computer plays a bit too defensively. We have tried to correct this with our attack and coordinated king attack sections, however, there is still room for improvement. Our end strategy section is still in development stages, but the general rules outlined above have been working well. The problem we are concentrating on now is a way of generalizing the various methods of checkmate, and how to press a material advantage. When these sections are completed we feel that they will greatly enhance our program's end strategy and overall playing ability.

Generalized principles for the computer to follow are quite difficult to program. However, in the initial way we have implemented this in our program at present, these principles have shown to be very effective in forming a consistent defensive strategy and a plan for offensive action. Experience with our program has shown them to be indispensible.

Our next article will analyze a sample game. We will have the opportunity to demonstrate that our program actually does what we say. In addition, we will give our outlook on the future of chess programming for the restricted user, and our views on how a home computer with limited space might be used to implement some of the major features of our program. We will not be able to publish a listing of our program since it would double the length of the magazine. Portions of the listing may appear.





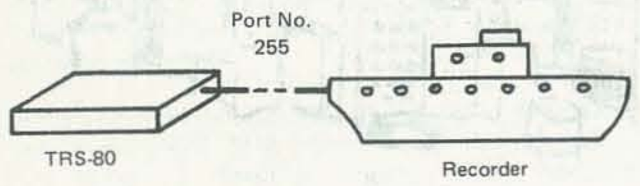
The Outside Connection
The Road Out

BY DOC PLUMBER AND SON, INK.

The "Plumber" family is growing. We now have Doc Plumber and Son showing you how to connect your TRS-80 to the outside world.

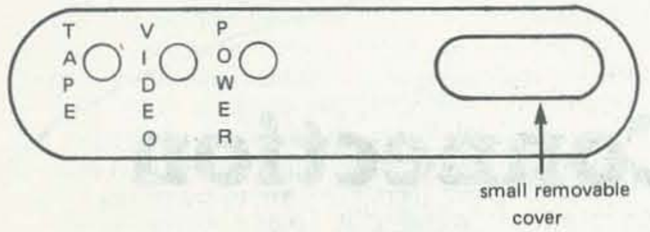
Doc (and Son) tell us this issue about an interface card that plugs directly into the back of the TRS-80 expansion interface port. In future issues, they will discuss particular applications they have developed that use the interface card. What's that noise? I believe I hear an electric train whistle . . . and a radio . . . and I smell fresh coffee! The Plumbers have a lot of surprises for us . . . look for them. – RZ

The TRS-80 is not the easiest computer to connect to the outside world, but there are ways to do it. I wrote about one of the ways (using the SOUNDWARE device) in the Nov.-Dec., Vol. 8, No. 3, 1979 issue of *RC*. That device plugged into the connection ordinarily used by the TRS-80's cassette recorder. The recorder itself sits outside the computer and makes its connection to the TRS-80 through a *port*. In other words, the recorder "docks" at the TRS-80 Port Number 255, the cassette plug-in point.



The cassette port is just one little connection and we can't hook up more than one device. This time I want to talk about how to connect several components to the TRS-80 at the same time. Where do we do this? We don't want to tear the TRS-80 apart to get at the electronic paths inside the computer. You probably know where we have to look – the rear of the keyboard.

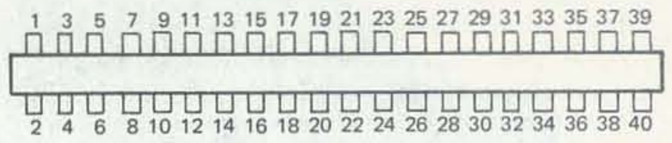
On the back of the keyboard is a small plastic cover that probably keeps falling off your unit. You may have even lost your's by now.



Rear View of the TRS-80

If you remove the cover, you will see a slot through which a portion of the printed circuit board (which holds most of the computer's electronic circuits) protrudes. This part of the printed circuit board is called an edge connector, for obvious reasons. It was designed for connection to the Radio Shack's Expansion Interface Unit. This unit is used to add memory and make connections to outside devices such as disk controllers and printers. The Expansion Interface is meant to be used for Radio Shack external products and is relatively expensive (approximately $300 without memory additions).

The edge connector contains 40 gold "fingers" (or contacts) which connect to the many paths (or lines) over which the computer's signals pass. These paths are usually referred to as the computer's bus (or buss, whichever you prefer).



TRS-80 40-Pin Edge Connector

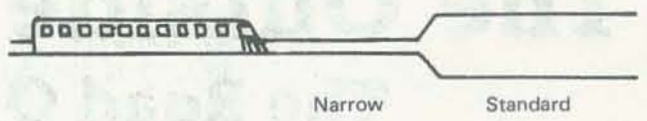
The computer's bus (or group of paths) carries address, data and control signals necessary for the operation of the computer.

With a suitable connector that matches the 40 gold fingers, you can tap into the TRS-80 bus and send signals back and forth. You can give the computer commands from your outside devices, and you can receive signals from the computer to control your devices.

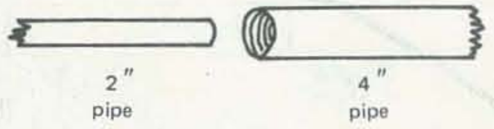
Let's not get too tied up in technical details. The important point is that you *can* connect your own external devices to the TRS-80. Some manufacturers (other than Radio Shack) are making it easy to do this.

Many products have appeared in the past that are used by computers which have a bus with 100 signal paths instead of 40. This system is known as the S-100 bus. These products include: graphic boards, memory boards, input/output boards, speech recognition and synthesis boards, and music boards. The list goes on and on.

To use all of these many useful products with the TRS-80, some method must be provided to make the 40 TRS-80 signal paths compatible with the 100 S-100 signal paths. It's like joining the tracks of a narrow gauge railroad with one of a standard gauge.



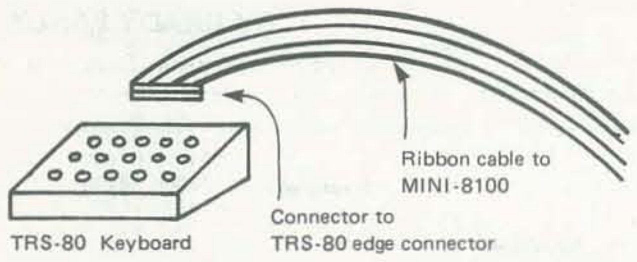
Or, as a Plumber might see it, it's similar to joining a 2 inch drain pipe to a 4 inch pipe. Only, it's not quite as easy.



In this article and future articles, I will be discussing one of the devices (called an interface) used to make the two nonmatching systems "fit."

HUH Electronics (acquired by California Computer Systems, 309 Laurelwood Road, Santa Clara, CA 95050) has produced several versions of a device which will perform this matching function. The version that we will be using is called the MINI-8100. It was purchased in kit form and put together by a 15 year-old high school student (the Son in Doc Plumber and Son, Ink.).

The MINI-8100 has a 40 pin connector with attached ribbon cable that connects to the back of the TRS-80 keyboard *or* to the Screen Printer Port of the Expansion Interface Unit of the TRS-80.



You must furnish a power supply for the MINI-8100 which in turn supplies the necessary power for the S-100 boards that you plug into the MINI-8100. There are provisions for 4 S-100 slots on the MINI-8100 board.

As you might imagine, there is no way to perfectly match the two systems. But the design of the MINI-8100 does a remarkably good job. *Do not* rush out and indiscriminately buy S-100 boards and expect them *all* to work on the MINI-8100. Their user's manual (provided with the kit) lists S-100 boards that have worked successfully. We will demonstrate the use of a few of these in future articles.

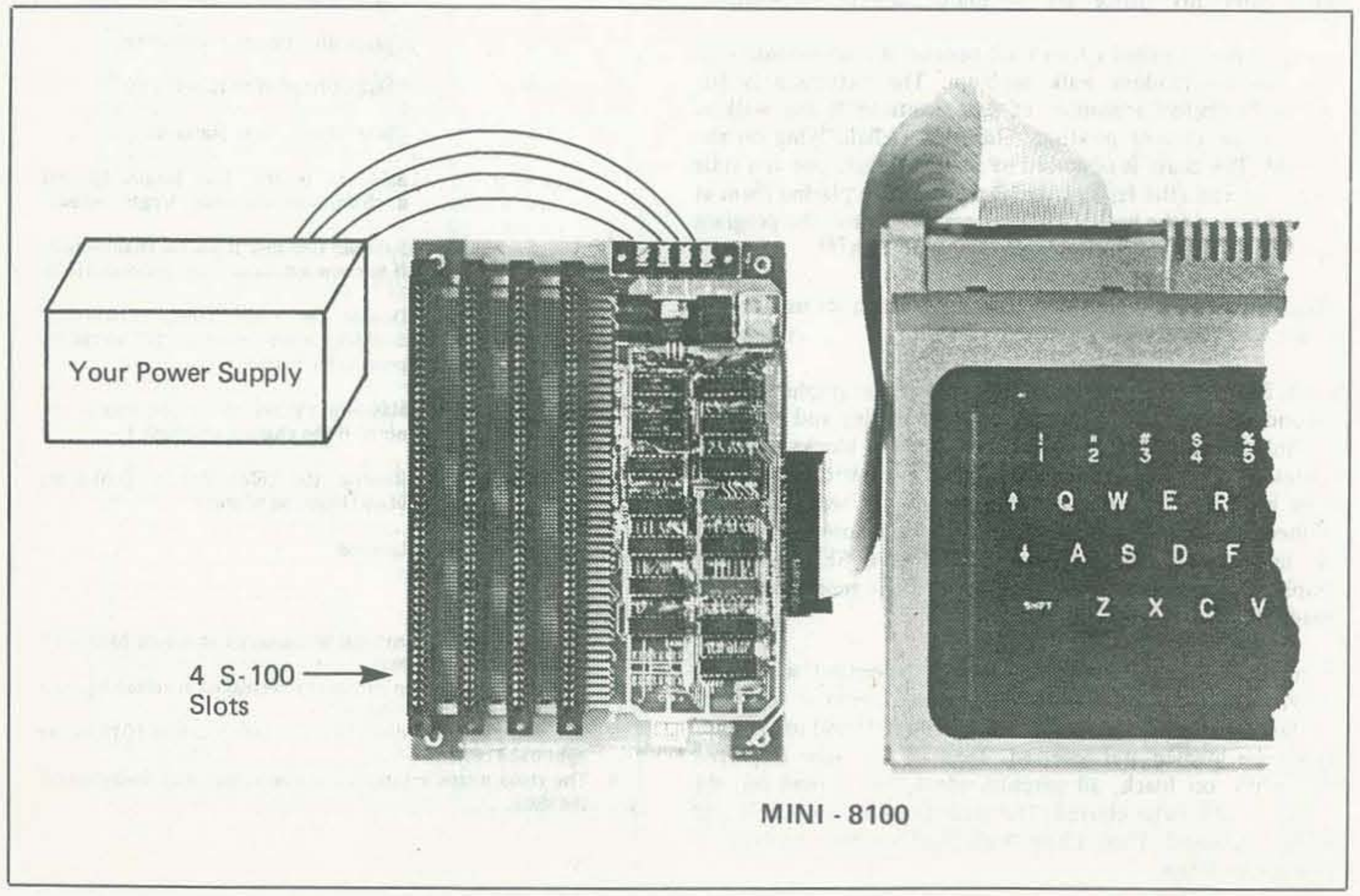
As the user's manual states:

Basically there are two restrictions: Any board which requires DMA (direct memory access) will not work and any board that requests a wait-state longer than one millisecond will also not work. The reason for both of these is that the dynamic memory refresh cycle (for the TRS-80's dynamic memories) will be interrupted for too long and the memory data will be lost.

So the two questions to ask of your prospective board supplier are: *Does this board use DMA?* and *Does this board request a wait state longer than one millisecond?* If the answer to any of these questions is yes, then the board will not work in the MINI-8100/TRS-80 system.

We'll discuss our first application in the next issue. If you have any ideas or projects that you would like to see demonstrated, let us know. The Plumbers are waiting.

more to come



New Ways to Use Set & Reset

TRS 80: Chain Walk

BY GRADY EARLY

Dr. Early is a professor in the mathematics department at Southwest Texas State University. He presents, in this article, an interesting graphical use of the TRS-80 that highlights some "common and not-so-common programming techniques."

Professor Early has been using computers since 1963, and currently is responsible for the Computer Science program at SWTSU in San Marcos, Texas. I can't stop myself from what's coming next... what happens to students who are late to Early's class? – RZ

When Chain Walk is running on my TRS-80 Level II, people who enter my office are fascinated–almost hypnotized.

The program is called Chain Walk because it is an extension of the classical random walk problem. The extension is the ability to display a number of past positions in the walk as well as the current position. Think of a chain lying on the ground. The chain is advanced by removing links one at a time from one end (the tail) of the chain and then placing them at the other end (the head). Under operator control, the program acts somewhat like an electronic Etch-a-Sketch™.

There are three notable programming techniques used in the Chain Walk algorithm.

First, SET and RESET are used to turn screen graphics blocks on and off, respectively, to simulate the adding and removing of links. The coordinates of the graphics blocks used to represent the chain links are stored in circular queues which have head and tail pointers. One step in the walk is accomplished by RESETing the graphics block whose coordinates are indicated by the tail pointer, and then SETting a new graphics block whose coordinates have been generated at the head pointer.

Second, the implementation of the I command was much easier than one might imagine. The memory locations containing the codes or tokens for SET (131) and for RESET (130) are located and reversed. Then, if the chain displayed was white on black, all graphics blocks are turned on; the screen is otherwise cleared. The walk proceeds with SET and RESET reversed. Thus, Chain Walk modifies itself to create a new screen image.

Features

Command (Keyboard entry)	Action
R	Random walk. No operator control is necessary.
C	Controlled walk. The walk continues in one direction until the operator changes direction. No 180 degree turns are allowed.
↑	Change direction to travel up.
↓	Change direction to travel down.
→	Change direction to travel right.
←	Change direction to travel left.
A	Add one to the chain length. Ignored if chain is at maximum length already.
S	Subtract one link from the chain length. If the new length is 0, the program stops.
D	Double the chain length. Ignored if doubling would exceed the maximum permissible length.
M	Massacre (halve) the chain length. Ignored if the chain is of length 1.
I	Reverse the video display (white on black; black on white).
All others	Ignored.

Other Features

1. The last command entered is displayed at screen location 0 (upper left hand corner).
2. The current direction of travel is displayed at screen location 960 (lower left hand corner).
3. The chain length is displayed at screen location 1015 (lower right hand corner).
4. The chain wraps around the screen rather than bouncing off the sides.

Third, any modification of the absolute memory locations referred to above requires knowing the exact locations of the tokens for the SET and RESET commands. This cannot be done *a priori* in a program intended for publication because the addition or deletion of even one single space by the person entering the program will alter the program memory map. This problem was solved by running a sequence of code which, when the program is RUN, will allow Chain Walk to find the appropriate memory locations (those containing a 130 or 131) and to remember them for later use in executing the I command. In effect, when Chain Walk wakes up (is run), it first orients itself with respect to memory and then proceeds. This orientation requires approximately 25 seconds. Notice in the program listing that there is no statement number 130; we would not want to confuse Chain Walk, would we?

Continued on pg. 54

LISTING

```
00010 CLEAR 100 : CLS
00020 DEFINT A-Z : DIM X(1000), Y(1000)
00030 I=1 : J=17129
00040 IF PEEK(J)=130 OR PEEK(J)=131 THEN 60
00050 J=J+1 : GO TO 40
00060 P(I)=J : I=I+1 : J=J+1 : IF I<6 THEN 40
00070 N=100 : MX=1000 : W=0 : DT=4 : H=1 : T=N
00080 FOR I=1 TO N : X(I)=N-I : Y(I)=0 : NEXT I
00090 XD(1)=0 : XD(2)=0 : XD(3)=-1 : XD(4)=1
00100 YD(1)=-1 : YD(2)=1 : YD(3)=0 : YD(4)=0
00110 FOR I=1 TO 4 : D$(I)=CHR$(90+I) : NEXT I
00120 RESET(X(T),Y(T)) : T=T-1 : IF T<1 THEN T=MX
00140 NH=H-1 : IF NH<1 THEN NH=MX
00150 X(NH)=X(H)+XD(DT) : Y(NH)=Y(H)+YD(DT) : H=NH
00160 IF X(H)>127 THEN X(H)=0 : GO TO 180
00170 IF X(H)<0 THEN X(H)=127
00180 IF Y(H)>47 THEN Y(H)=0 : GO TO 200
00190 IF Y(H)<0 THEN Y(H)=47
00200 SET(X(H),Y(H)) : D$=INKEY$
00210 IF D$="↑" THEN ND=1 : GO TO 260
00220 IF D$=CHR$(10) THEN ND=2 : GO TO 260
00230 IF D$=CHR$(8) THEN ND=3 : GO TO 260
00240 IF D$=CHR$(9) THEN ND=4 : GO TO 260
00250 GO TO 270
00260 D$=D$(ND) : PRINT @0,D$; : GO TO 420
00270 PRINT @0,D$; : PRINT @960,D$(DT); : PRINT @1015,N;
00280 IF D$="" AND W=0 THEN 120
00290 IF D$="" THEN ND=RND(4) : GO TO 420
00300 IF D$="I" THEN 560
00310 IF D$="M" AND N>1 THEN 440
00320 IF D$="D" AND N<MX/2+1 THEN 470
00330 IF D$="R" THEN W=1 : GO TO 120
00340 IF D$="C" THEN W=0 : GO TO 120
00350 IF D$<>"A" THEN 370
00360 IF N=MX THEN 120 ELSE N=N+1 : GO TO 140
00370 IF D$<>"S" THEN 410
00380 IF N=1 THEN CLS : STOP
00390 RESET(X(T),Y(T)) : N=N-1 : T=T-1 : IF T<1 THEN T=MX
00400 GO TO 120
00410 IF W=0 THEN 120 ELSE ND=RND(4)
00420 IF ND=DT OR DT+ND=3 OR DT+ND=7 THEN 120
00430 DT=ND : GO TO 120
00440 ML=N/2 : N=N-ML
00450 FOR I=1 TO ML : RESET(X(T),Y(T)) : T=T-1 : IF T<1 THEN T=MX
00460 NEXT I : GO TO 120
00470 ML=N : XD=XD(DT) : YD=YD(DT) : N=N+ML
00480 FOR I=1 TO ML : NH=H-1 : IF NH<1 THEN NH=MX
00490 X(NH)=X(H)+XD : Y(NH)=Y(H)+YD : H=NH
00500 IF X(H)>127 THEN X(H)=0 : GO TO 520
00510 IF X(H)<0 THEN X(H)=127
00520 IF Y(H)>47 THEN Y(H)=0 : GO TO 540
00530 IF Y(H)<0 THEN Y(H)=47
00540 SET(X(H),Y(H)) : NEXT I
00550 GO TO 120
00560 I1=130 : IF PEEK(P(1))=130 THEN I1=131
00570 I2=261-I1
00580 POKE P(1),I1 : POKE P(3),I1 : POKE P(4),I1
00590 POKE P(2),I2 : POKE P(5),I2
00600 IF I1=130 THEN CLS : GO TO 120
00610 FOR I=1 TO 16 : PRINT STRING$(64,CHR$(191)); : NEXT I
00620 GO TO 120 : END
```

Set character storage, clear screen, type variables, dimension large arrays.

Find memory locations containing the SET and RESET tokens.

Initialize simple variables.
Initialize queues.

Initialize increment arrays and arrow character array.

Move one step in current direction, DT. Wrap around screen if necessary. Strobe keyboard for new command, D$, if any.

If command was an arrow, convert from cursor control codes to character display codes.

Print screen corner data; D$, D$(D), N.

Check for null, I, M, D, R or C commands and transfer control accordingly.

Service A command.

Service S command.

Determine new value for DT (direction for next step).

Service M command.

Service D command.

Service I command.

this publication is available in microform

Please send me additional information.

University Microfilms International

300 North Zeeb Road
Dept. P.R.
Ann Arbor, MI 48106
U.S.A.

18 Bedford Row
Dept. P.R.
London, WC1R 4EJ
England

Name____________________
Institution____________________
Street____________________
City____________________
State__________ Zip__________

Continued from pg. 53

Data Definitions

Variable	Usage
N	Number of links in the chain; initialized at 100.
MX	Maximum number of links allowed; = 1000.
X, Y	Circular queues for link coordinates.
XD, YD	Arrays of X, Y increments used to generate new head link coordinates.
DT	Direction of Travel (current). 1 = up, 2 = down, 3 = left, 4 = right. DT = 4 is initial direction.
ND	Proposed new direction of travel. Cannot be 180 degree reversal.
P	Array of memory locations containing the SET and RESET tokens.
H, T	Index of head and tail, respectively, of chain in the X, Y queues.
W	Set to 0 for controlled walk; to 1 for random walk.
D$	Current command (scalar variable); Codes for arrows (array variable).
ML	Temporary variable used to indicate the number of links to be added or deleted during execution of the D or M commands.
I1, I2	Values of tokens (130 and 131) used to reverse SET and RESET during execution of an I command.

Overpopulation, inflation, fuel shortages, poverty, hunger. The problems of the world can be overwhelming, especially when there is no end in sight. But now there is a solution.

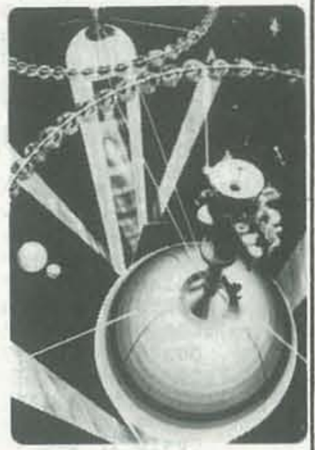
COLONIZE OUTER SPACE

A century's progress in transportation.

Solar Power Satellite

There is a group of people working to mold the future of humankind —a future they believe lies off the surface of the planet.....in outer space.

The **L-5 News** provides information about the latest developments in the aerospace industry as well as discerning comments about how they could affect your future.


Space colony drawn from actual designs.

L5

Your $20 membership includes a subscription to the **L-5 News,** merchandise discounts, mailings from the Legislative Information Service, and a brighter looking tomorrow.

**Please send orders to: L-5 Society
1060 E. Elm St.
Tucson, AZ 85719**

Reviews

TEMPLE OF APSHAI
By Automated Simulations
P. O. Box 4232
Mountain View, CA 94040
$24.95

This game is one in the DUNJONQUEST series by Automated Simulations. It is a solitare Dungeon and Dragon type simulation game, with the PET acting as the dungeon master. In this game, the player controls or is a character that moves through a series of rooms searching for treasure, somewhat as in ADVENTURE, but with *much* more complexity involved. You are given a character at the beginning with 6 familiar attributes–Intelligence, Intuition, Ego, Strength, Constitution, and Dexterity–or you can design your own. You then have your choice of 4 levels of increasing difficulty. Each level has 56 to 60 rooms on it, with 23 different Monsters possible, a total of 80 treasures, and a whole raft of various traps, magical items, hidden doors, and other items.

A plan of the room is drawn on the screen of the PET, showing treasures, walls, doors, and Monsters (if any). The screen is animated, and drawn in large scale. Watching the PET draw the rooms is entertainment in itself! There are 18 commands available.

The complexity of this game is incredible! You can keep playing this game literally for days before you explore each room and get all the Treasures.

For $24.95 you get a tape (program on one side, a data tape for the 4 levels on the other side) and a 56 page booklet. The instruction booklet is very well written, and is the best I have ever seen for a computer game. The operating program occupies 24K, and the data occupies another 6.5K, so only a 32K PET can run this game.

There are a few criticisms, though. The first part of the operating program could be written to display the INPUT requests in a more attractive manner; it's strictly a teletype-style display as is. It takes a *LONG* time to load via the cassette deck. Commodore's speedy 500 BPS tape deck is to blame for this. This program *begs* for a floppy disk!

The instructions for loading state: "Although the program is designed for the new PET, you should experience no difficulty in running it on an old, expanded PET, other than reversed upper and lower case characters." This statement is totally, absolutely and completely *wrong!* A Level I PET has nothing *but* problems, some of them *very* weird, and beyond understanding. If you change *all* the upper/lower case characters, some INPUT statements won't accept *anything*! At least one INPUT A$ statement requires the string to be entered *twice*. On the second request for A$ (after it prints "??"), if you RETURN without giving it any input, the PET will return A$ = 0! Under normal circumstances, this would cause the PET to drop out of BASIC. I relocated two INPUT lines to the end of the program, and went GOSUB to them, and that cured those problems.

There's a ton of INT(RND(0)) statements in this program. In the new PETs this is O.K. In the old PETs, this is a No-No! Level I RND(0) always returns the same number, so you find yourself playing exactly the same game over and over again. You must go through the *entire* program, and change *all* those 0s (about 30) to a positive value. A bigger problem is the dice-rolling subroutine in line 2190. This simulates throwing 3 dice to get a number between 3 and 18. After fooling around for about 2 hours, I finally came up with a fix:

Replace line 2190 with:

```
2190    J=INT(6*RND(TI))+INT(RND
        (-TI) )+1
2191    J=J+INT(6*RND(TI))+INT
        (RND(-TI))+1
2192    J=J+INT(6*RND(TI))+INT
        (RND(-TI))+1
2193    RETURN
```

Doing any debugging or modification is a real pain, due to the programmer's style, and the length and complexity of this program. There's a jillion GOSUBs, GOTOs, IF-THENs, DIMs, POKES, FOR loops, and an incredible number of variables.

On the positive side, this program runs OK in a friend's Level II PET. It appears that the authors never tried it out on a Level I PET, and assumed that the upper/lower case character difference was the only change needed, as many people do. Automated Simulations could make more data tapes for more levels, and sell them at a lower price (hopefully) to addicts like me, who already own the basic game and instruction book. I'd buy them all!

This is *not* a game for the 8-years-olds to play, because the rules are complex, and it takes several hours to play a meaningful round of Dunjonquest. A weekend can pass unnoticed when you get involved in this game.

I wholly recommend this game, because it is a well designed, absorbing and exciting game. A masterpiece of Level II programming. *WARNING! Extremely* addictive and time-consuming. Not for the faint of heart or married. (My wife *HATES* this game!)

Reviewed by David Conley
Santee, CA

PROGRAMMING PROBLEMS

BY BOB ALBRECHT AND DON ALBERS

A new feature for the magazine! Problems for you to ponder, puzzle over, program, and take pleasure in solving. The Dragon (Bob Albrecht) and Don Albers promise that this section will become the number one source of programming problems for teachers. They encourage teachers to use these problems in the classroom—to have students solve the problems and send in solutions for publication. Of course, we encourage all of you to give the problems a try. Some are easy, some are really difficult. Let's hear from everyone on this new set of challenges! – *RZ*

We begin an ongoing series of programming problems for learners, teachers, or anyone who likes to solve problems. Our problems will range from easy, to harder, to awful. They will be numbered 1, 2, 3, 4, 5 and so on. In future issues, solutions to previously published problems will appear —we hope that *your* solutions will appear. We also hope that you will send problems for us to publish.

Some of the problems in this series (we won't tell you which ones!) will be used as subjects of "mini-tutorials" in future issues. These tutorials will be designed to help break "mind-sets" or "mental-roadblocks" or "computerized conventional wisdom" by showing unusual ways to solve problems. We hope that *you* will send unusual solutions. We especially want solutions that make people say, "That's so simple (beautiful, elegant, etc.). Why didn't *I* think of that?"

And now, on to the first bunch of problems. Send your solutions to these problems; also send problems.

PROBLEM #1 POSITIVE, NEGATIVE OR ZERO

An easy problem. All we want is a program that asks for a number, then tells you something about the number.

- If you enter a positive number, the computer tells you: YOUR NUMBER IS POSITIVE
- If you enter a negative number, the computer tells you: YOUR NUMBER IS NEGATIVE
- If you enter the number zero (0), the computer tells you: YOUR NUMBER IS ZERO

A solution to this problem is shown below, in TRS-80 BASIC, a version of Microsoft^{TM} BASIC. With minor editing it will run on the APPLE, ATARI, PET and several other computers.

```
100  REM***PROBLEM  #1 POSITIVE, NEGATIVE OR ZERO
110  REM***RECREATIONAL COMPUTING, JAN/FEB 1980
120  CLS

300  REM***ASK FOR A NUMBER, X
310  PRINT : INPUT "NUMBER, PLEASE" ; X

500  REM***TELL WHETHER NUMBER IS POSITIVE,
     NEGATIVE OR ZERO
510  IF X > 0 THEN PRINT "YOUR NUMBER IS POSITIVE"
520  IF X < 0 THEN PRINT "YOUR NUMBER IS NEGATIVE"
530  IF X = 0 THEN PRINT "YOUR NUMBER IS ZERO"

700  REM***GOTO 'ASK FOR A NUMBER, X'
710  GOTO 310

999  END
```

Ha! We have pre-empted the solution which immediately leaped to your mind. In fact, you *can't* use the IF statement at all! *Your* program must "behave" exactly as ours, but without any IF statements.

And yes, old timers, BASIC experts and others, we know that there is an obvious way to solve this problem, using a statement that begins with "O" and a function that begins with "S". If you do it that way, you will be one of many. So . . . rethink! We gleefully await your *unusual* solutions.

PROBLEM #2 PALINDROME NUMERALS

These are palindromes: 121, 747, 606, 12321, 3773
These aren't palindromes: 123, 76, 5049, 37, 3774

A palindrome is something that reads the same from right to left as it does from left to right.

'23' is not a palindrome!

But '23' is the decimal numeral for the number whose English name is 'twenty three.' Hmmm . . . that number is called 'drei und zwanzeig' in German. It is also called '212' in *base 3*.

'212' (base 3) is a palindrome!

Aha! A number is a number, but the *name* of a number may or may not be a palindrome.

1. 27 (base 10) is not a palindrome.
   27 (base 10) = 11011 (base 2) = 33 (base 8)
   '11011' and '33' are palindromes.

2. 121 (base 10) is a palindrome.
   121 (base 10) = 11111 (base 3) = 232 (base 7)
   '121', '11111' and '232' are palindromes.

Write a program to:

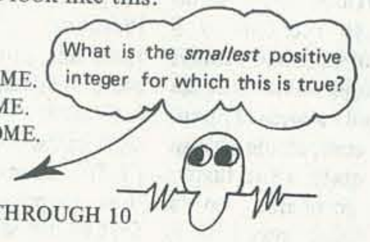
1. Ask for the decimal name of an integer in the range 1 to 999999. (Reject all other numbers)

2. Show all bases 2 through 10 in which the numeral for the number entered in (1) is a palindrome. Also show the numeral in each base.

3. If there is *no* palindrome numeral in a base 2 through 10 for the number entered in (1), show the message: NO PALINDROME IN BASES 2 THROUGH 10.

A RUN of your program might look like this.

```
YOUR NUMBER?  121
11111  (BASE 2) IS A PALINDROME.
232    (BASE 2) IS A PALINDROME.
121    (BASE 10) IS A PALINDROME.

YOUR NUMBER? 1234
NO PALINDROMES IN BASES 2 THROUGH 10

YOUR NUMBER? 1000000
SORRY. I PLAY ONLY WITH INTEGERS, 1 TO 999999.

YOUR NUMBER? 3.14159
SORRY. I PLAY ONLY WITH INTEGERS, 1 TO 999999.

YOUR NUMBER?  and so on.
```


What is the *smallest* positive integer for which this is true?

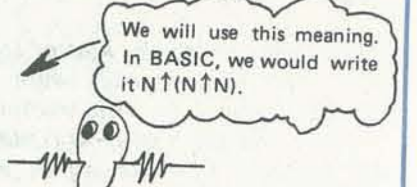
Acknowledgement: This problem and the one that follows are paraphrases of similar problems in *Advanced Problems for Computer Mathematics* by Bob Albrecht, published by Digital Equipment Corporation.

PROBLEM #3(a) N TO THE N TO THE N

If N is a positive integer then N^{N^N} is also a positive integer.

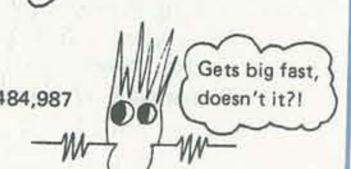
Hmmm . . . we can read N^{N^N} in two ways.

1. N^N to the N = $(N^N)^N$
2. N to the N^N = $N^{(N^N)}$


We will use this meaning. In BASIC, we would write it N↑(N↑N).

Here are some examples:

1. $1^{1^1} = 1^{(1^1)} = 1^1 = 1$
2. $2^{2^2} = 2^{(2^2)} = 2^4 = 16$
3. $3^{3^3} = 3^{(3^3)} = 3^{27} = 7{,}625{,}597{,}484{,}987$
4. $4^{4^4} = 4^{(4^4)} = 4^{256} = ???$



That's problem # 3(a). Compute the exact value of 4^{4^4}. It can be written as a decimal numeral with 155 digits. If you wish, try 5^{5^5} or 6^{6^6} or . . . .

PROBLEM #3(b) HOW MANY DIGITS IN N^{N^N}

Here comes problem 3(b). How many digits does the *expanded* form of N^{N^N} have, if the number is expanded in decimal?

1. $1^{1^1} = 1$	The expanded form has 1 digit.
2. $2^{2^2} = 16$	The expanded form has 2 digits.
3. $3^{3^3} = 7{,}625{,}597{,}484{,}987$	The expanded form has 13 digits.

4. The expanded form of 4^{4^4} has 155 digits.
5. The expanded form of 5^{5^5} has 2185 digits. Hmmm . . . does the expanded form of 6^{6^6} have too many digits to be stored in the memory of a home computer?

Write a program to:

1. Ask for a positive integer (N) in the range 1 to 9.
2. Show the number of digits in the expanded form of N^{N^N}.

Quickly, without using a computer—how many digits in the expanded form of $10^{10^{10}}$?

PROBLEM #4 VOLUME OF A POTATO

We hand you a potato. We want the volume of the potato in cubic centimeters. It could be any old (or new) potato. What do you do? How can your computer help?

The saga of the potato will continue in *RC*. Please send your ideas—given a potato and a computer, how can we obtain the volume of the potato?

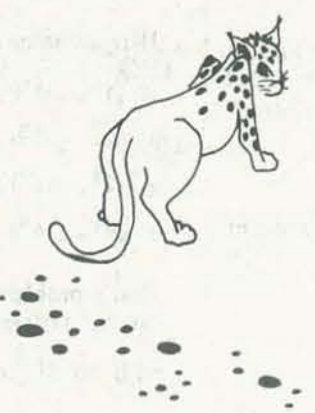
SPOT

The Society of PET Owners and Trainers

BY HARRY J. SAAL

Commodore's PET is a factory-assembled personal computer based on a 6502 microprocessor. The original PET, model 2001-8, was a $795 system that included a keyboard, cassette tape unit, built-in TV screen, some graphics, upper and lower case, extended 8K BASIC, and 8K of user memory.

SPOT is devoted to the host of applications–routine and wild–which PET users have found for their machines, as well as to the nitty-gritty of repairs and modifications. In other words, almost anything relating to the PET is fit material for this column. Just send Harry your questions, ideas, and tapes c/o PCC. He'll give each of them his careful attention. – *TD*



HEARD AROUND THE QUAYSIDE

Commodore continues to make changes in its product line. The latest is the discontinuance of the original 2001-8 machine, with the built-in cassette tape and small keyboard. In its place is the 2001-8N, which, except for memory size, is identical to its bigger brothers that have 16 or 32K memory. In effect, it means a price increase for economy-minded PET purchasers, since it costs an additional $100 to get a tape cassette. What you do get is the full-size keyboard, and the new corrected ROM set.

Speaking of price changes, Commodore has twice now made a super offer to school systems, given them three machines for the price of two. But recently, the offer was abruptly shortened by 30 days. This caused havoc with many PET dealers, who planned extensive advertising and sales campaigns based on Commodore's offer. It is stunts like this that will cause the PET to lose the backing of its army of supporters.

Several other changes seem to be in the wind. Commodore is getting ready to release a new enclosure for the PET, made of structural foam (like the Apple) rather than steel. The appearance of the PET is somewhat "softened," but otherwise unchanged. The PET and TI 99/4 were the only two personal computers tested by the FCC that meet the July 1980 limits on RF interference. Let's hope this change doesn't affect the excellent shielding currently provided by the metal case. Rumor also has it that Commodore will be switching the character generator on the PET *back* to the way it was on the first 8K PETs. It was a mistake to change it in the first place, but this will throw even more confusion into the arena!

Lastly, Commodore is hard at work on a "business" machine. The major change is the use of a larger 80 character screen, and built-in floppy disks. (Sound like the Tandy TRS-80 Model II??). Will it have a 6502 CPU, or a 6809 as several of the newer products now under development elsewhere are expected to have? The 6809 is a nice upgrade from the 6502. It contains a series of 16 bit extensions to the current 8 bit architecture of the 6500/6800 family, but to the outside world looks like an 8 bit processor, so all standard peripheral chips are still compatible. Compared to a 6800 it is quite a bit faster; unfortunately when compared to a 6502 running at the same speed it is pretty much even, although easier to write programs for.

PET TIPS

You may have noticed that on the new PET even when you POKE characters to the screen there is no "snow." This same change permits the PET to print characters or list programs much faster than before, although Commodore's software doesn't take advantage of it. But you can make your PET go faster by issuing a POKE 59458,62. Now LIST a long program! (To restore the PET back to the normal mode, issue POKE 59458,30.) Doing this on an 8K PET also makes things go faster, but the screen blinks on and off in a very disturbing fashion.

Did you know that the machine language monitor supplied by Commodore is in ROM on the new Pets? The manual indicates that you load it from tape, but it's really there. One quick way to start it up is to SYS(1024). Details about using the monitor are in the User's Manual; remember that an X command gets you back to BASIC.

We all know that programs which do PEEKs and POKEs often work on one model of the PET, but not on the other. I was surprised to see a program which worked fine on an 8K PET have a totally messed up display format when run on a new PET, and there wasn't a PEEK or POKE to be found. There are changes in *BASIC* itself to be wary of! One example is this little expression: typing ?SPC (40);:?POS(0) on an old PET prints 40, and on the new PET prints 0. What changes have you run into?

PET PUBLICATIONS

The PET Gazette is no longer. Len Lindsay's fine magazine has disappeared, but is replaced by *COMPUTE*, The Journal for Progressive Computing. Issue 1, dated Fall 1979, is already out. It is a beautiful, slick 102 page magazine with lots of good information and sources of products for the PET. Robert Lock, President of Small System Services, Inc., has put together a superb replacement magazine, and has Len Lindsay as his Senior Contributing Editor. As a sign of how much work went into *COMPUTE*, someone managed to get Commodore to pay for 4 pages of advertising, a first to my knowledge.

There is a change in orientation from the PET Gazette. More like MICRO, *COMPUTE* plans to provide information for all 6502 based machines, which includes the Apple, KIM, SYM, AIM, OSI and the new Atari. You can subscribe to The Journal for $9 for six issues (one year). An alternative is a $7.50 "personal/retail" subscription, where your copy is bulk delivered to a local dealer for you to pick up, instead of being sent by Pony Express. Sounds like a cheaper and faster way to get *COMPUTE*. Subscriptions can be ordered from *COMPUTE*, P.O. Box 5119, Greensboro, NC 27403. All technical correspondence can still be sent to Len Lindsay, 1929 Northport Dr. #6, Madison, WI 53704.

WORKBOOKS

There are two new student workbooks for beginners learning PET BASIC. Both are in active classroom use, and are designed to be used with the PET at hand, yet don't need an instructor nearby. They gently and effectively introduce the PET, starting with the keyboard. Cow Bay Computing, Box 515, Manhasset, N.Y. 11030 has released *Feed me, I'm Your PET Computer* for beginners. It is available for $4.95, and has a companion book *Looking Good With Your PET* for intermediates (also $4.95).

A series of similar workbooks was developed at San Jose State University, with parallel editions for the PET, Apple and TRS-80. Beginners can get started with *Training your Computer (PET edition)*. Ten or more copies can be ordered from METRA Instruments, Inc., Pickering Division, 2056 Bering Drive, San Jose, CA 95131, for $2 each. Single copies may be ordered from Creative Publications, P.O. Box 10328, Palo Alto, CA 94303, for $4 a copy.

PET INTERFACES

If you are interested in interfacing any devices to your PET through the user port, or in understanding how the CB2 music works, you probably should take a look at Rodnay Zaks' "6502 Applications Book." It is published by SYBEX, 2020 Milvia Street, Berkeley, CA 94704. This book is chock full of details on all the various peripheral chips that are used in the PET, and includes details about programming them in assembly language. The book costs $12.95 and is available in most computer retail stores.

PET HARDWARE ADD-ONS

One of the more remarkable gadgets to put on a PET has been developed by Innovision, P.O. Box 1317, Los Altos, CA 94022. The Prestodigitizer^{tm} tablet attaches to the PET's user port, and can be used to provide handwritten input to the PET. (See Prestodigitizer article by David Thornberg, this issue). Sample programs which come with the Prestodigitizer tablet can recognize all the numbers or letters when written on the tablet using the metal tipped stylus provided. In fact, you can even train the PET to recognize your own handwriting. This handsomely packaged device comes with a useful manual, and cassette tape with several sample applications. It costs $48.50 plus $1.50 (and tax in California), and can be ordered either from Innovision, or at local computer dealers.

Have you put sound on your PET yet? If not, the best way to go is to get the Soundware adapter, from CAP Electronics, 1884 Schulman Avenue, San Jose, CA 95124, or your local retailer, for $29.95. It's very attractively packaged, and includes a handy instruction book and demo tape. Even if you already made your own "CB2" sound box, get a hold of the tapes from Soundware called Action Pack, The Classics, and Word Fun. Each costs only $9.95, and contains three excellent programs with sound incorporated. They are well done, and quite inexpensive.

Other sources of interesting music tapes (aside from many of the fine programs still coming regularly from *CURSOR* Magazine) are the Allen Computer Products Music Box 1 and Animation 1 tapes. These two tapes are available at $10 each from Allen Computer Products, 34844 Munger Drive, Livonia, MI 48154. Each contains four selections from Bach, Chopin and Mozart including the Can-Can and Flight of the Bumblebee. All eight programs have amusing animated screen displays while the music plays. Unfortunately the author's attempt to make the programs run on either old or new ROM machines is somewhat muddled and inconsistent. Just the same, they contain quite a bit of pleasant whimsy, although the quality is not as high as those from Soundware or *CURSOR*.

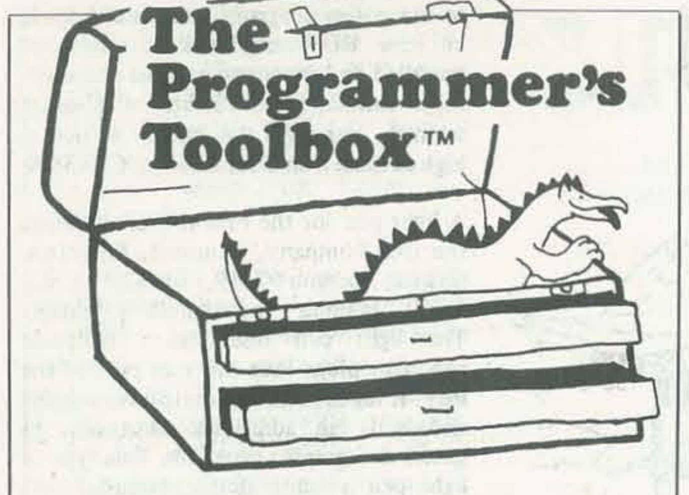
A light pen for the PET is available from the 3G Company, Route 3, Box 28A, Gaston, Oregon 97119, for $29.95 plus $1.50 mailing and handling charge. The light pen resembles a ballpoint pen, and plugs into the user port of the PET. It requires no external power supply, and adds an additional dimension to games and graphic programs. This type of light pen is quite simple, essentially responding to the presence or absence of light. By displaying one or more flashing cursors on the screen, and testing in software whether the blinking of the cursor is in sync with what the light pen sees, it is simple to set up multiple choice or menu driven programs. The light pen I tested did not have polarizing plugs on the user port connector, and the orientation of the connector always seemed "upsidedown" to me until I placed a "this side up" sticker on the back. Other than this oversight, it is a very well designed product, and one that adds a lot of enjoyment to the PET. Software which is written for use by this light pen is available from several sources. I found the programs written by Quill Software, 2512 Roblar Lane, Santa Clara, CA 95051 to be the best. The light pen is best suited for a multiple choice scenario, such as in the game of Swords and Sorcery, Hunt the Wumpus, or Othello. There are four tapes avilable from Quill, priced at about $20 each. I suggest you write them for a complete up-to-date list as new ones are made available regularly.

CORRECTION

In the Sept-Oct issue of *RC*, Harry Saal wrote that the Metagame *Hunt* was available from Computer Way in Madison, WI. However, the author of the game, Michael Richter, informed us that it is really available from:

Programma International, Inc.
3400 Wilshire Blvd.
Los Angeles, CA 90010
(213) 384-0579

The Programmer's Toolbox™



BY EVERYBODY

In Vol. 1, No. 3 of *PC*, 1973, Marc LeBrun began a column that provided routines that could be used as part of a "toolbox" of computer skills. We revived that column in the May-June 1979 issue of *RC*.

Here are some more "tools" for the Toolbox. Hope you can use these new programs. If you have ideas for useful routines, write them down and send them to us. —RZ

PT11: MENU SELECTOR

Here is a more elegant form of a "menu selector" than the usual "Enter the number of the function desired". This routine accepts any single character, compares it to an arbitrary list of characters, and returns with a number which corresponds to the matched character. The return value can be used in an ON . . . GOTO . . . statement. If there is no match, a nicely formatted list of the acceptable characters is printed. The routine accepts lower case letters. Written as a multiple line defined function in Northstar BASIC, the function can be easily converted to a simple subroutine or to other versions of BASIC.

```
1000 DEF FNI(B$)
1010   T=LEN(B$)
1020   I=0\A$=INCHAR$(0)\IF A$=CHR$(13) THEN 1140
1030   IF A$<" " THEN 1020 ELSE PRINT A$
1040   IF A$=>"a" THEN A$=CHR$(ASC(A$)-32)
1050   FOR I=1 TO T
1060     IF A$=B$(I,I) THEN EXIT 1150
1070   NEXT I
1080   PRINT "ENTER ONLY ",
1090   FOR I=1 TO T
1100     PRINT B$(I,I),
1110     IF I<T-1 THEN PRINT ", ",
1120     IF I=T-1 THEN PRINT " OR ",
1130   NEXT I\PRINT ": ",\GOTO 1020
1140   PRINT
1150   RETURN I+1
1160 FNEND
```

Line 1020 inputs the character and checks if it is a carriage return.
Line 1030 checks for control codes (which are accepted by the INCHAR$ function) and ignores them.
Line 1040 converts lower case letters to upper case.
Lines 1050 to 1070 check for a character match. Jumps to line 1150 if a match is found.
Lines 1080 to 1130 are reached only if no match was found. They print out a list of the acceptable characters.
Line 1140 is reached only if the input was a carriage return.
Line 1150 returns the value of this function: 1 if it is a carriage return, otherwise the number of the matched character + 1.

Finally, a few words on Northstar BASIC:

The \ is the statement separator, equivalent to : in other BASICs.
Line 1020: A$ = INCHAR$(0) puts the next key pressed into A$, it does not print it. It will accept control codes.
Lines 1060 and 1100: B$(I,I) gets the Ith character alone from B$. It is equivalent to MID$(B$, I, 1).
Line 1060: Northstar BASIC requires EXIT to prematurely terminate a FOR/NEXT loop. Other BASICs just use GOTO.
The comma in PRINT statements is the same as a semicolon in other BASICs.

BY LARRY HUDSON

PT12: FACTORIAL ROUTINE

This function and the one following, in PT13, are two more Northstar BASIC routines. For a short and simple way to compute factorials, give this one a try. Refer to PT11 for information on how Northstar BASIC corresponds to a lot of standard BASICs. Warning: this routine is *recursive*—it calls itself in line 40!

```
10 REM FACTORIAL FUNCTION USING NORTHSTAR BASIC
20 DEF FNA(X)
30 IF X>1 THEN 40 ELSE Y=1\RETURN Y
40 Y=FNA(X-1)*X
50 RETURN Y
60 FNEND
70 INPUT X\PRINT #1,X,"!= ",FNA(X)
80 END
 1!=  1
 5!=  120
 10!=  3628800
 20!=  2.432902E+18
 49!=  6.0828188E+62
```

BY R.BLESSING

PT 13: ACKERMANN'S FUNCTION

For those of you with a need to compute a complex function that requires a routine to call upon itself, here is a fine example. In lines 40 and 50, the function calls itself to continue the calculations. Line 50 is especially interesting. Some BASICs may not handle this kind of *recursive* operation, so programmers beware!

```
10 REM ACKERMANN'S FUNCTION USING NORTHSTAR BASIC
20 DEF FNA(A1,A2)
30 IF A2<>0 THEN 40 ELSE Y=A1+1\RETURN Y
40 IF A1<>0 THEN 50 ELSE Y=FNA(1,(A2-1))\RETURN Y
50 Y=FNA((FNA((A1-1),A2)),A2-1)
60 RETURN Y
70 FNEND
80 INPUT "A1 & A2 ",A1,A2\PRINT #1,"A1= ",A1,"  A2= ",A2
90 !#1\!#1
100 FOR I=0 TO A2
110 FOR J=0 TO A1
120 PRINT #1,TAB(J*6),FNA(J,I),
130 NEXT J
140 PRINT #1
150 NEXT I
160 END

A1=  9  A2=  3


 1     2     3     4     5     6     7     8     9     10
 2     3     4     5     6     7     8     9     10    11
 3     5     7     9     11    13    15    17    19    21
 5     13    29    61    125   253
```

BY R. BLESSING

PT14: NAME GENERATOR

As part of a new game, called EARTHQUEST, a routine was needed to generate exotic names of places and objects. The variable NL is set to values of 2, 4, 6, or 8. C$ contains the string "BCDFGHJKLMNPRSTVWYZ". V$ contains "AEIOU". The routine avoids obscenities in English, but persons who speak Old High Martian may be offended.

```
1000 REM  PICK NAME
1010 FOR I=1 TO NL STEP 2
1020 C= RND (19)+1
1030 N$(I)=C$(C,C)
1040 V= RND (5)+1
1050 N$(I+1)=V$(V,V)
1060 NEXT I
1070 RETURN
```

BY JIM DAY

PT15: LPRINTING BY POKEING

There is an easy way to direct material to the line printer, and I suspect it will work for your Quick Printer as well. I can't take the credit for discovering them, but they have helped me to write programs that can be run on systems with and without line printers.

Whenever you want to send data to the printer, use the following:

POKE 16414,141 : POKE 16415,5

Whenever you want to restore sending data to the screen, use this:

POKE 16414,88 : POKE 16415,4

When the first instruction set is executed, the computer thinks that the line printer is the screen, so PRINT statements act like LPRINT statements. The second set of statements restores the screen as the output device. When you use the regular line printer, the computer can tell if the printer is ready to receive data:

IF PEEK (14312) = 63 THEN POKE 16414,141 : POKE 16415,5

This last operation may not work with the Quick Printer.

Also, when you return to the screen, the first character in the next PRINT statement is lost, so always print a null ("") or a blank line, as well as CLS the screen just before executing the first instruction. Finally, don't try to input data while using the line printer as above, since the prompt won't appear on the paper until after you enter the data.

BY MILAN CHEPKO

PT 16: TRS-80 GRAPHICS CHARACTERS

This program will put all the TRS-80 graphics on your screen, side by side with the ASCII codes (129 through 191) for each character. You will see the graphics characters, eight per line, double-spaced on the screen so that you can easily distinguish one character from another.

```
100 REM ***TRS-80 GRAPHICS CODES AND CHARACTERS
110 REM ***PROGRAMMER'S TOOLBOX (TM) #16
120 REM ***RECREATIONAL COMPUTING, JAN/FEB 1980
130 CLS

200 REM ***SHOW CODES AND CHARACTERS ON THE SCREEN
210 FOR CODE = 129 TO 191
220 PRINT CODE CHR$(CODE) STRING$ (2,32);
230 IF POS(0) = 0 THEN PRINT
240 NEXT CODE

300 REM *** WAIT WHILE ENRAPTURED WATCHER WATCHES
310 GOTO 310

999 END
```

Line 220 prints exactly eight (8) characters each time it is executed, as shown below.

$$\underbrace{\text{220 PRINT CODE}}_{\substack{\text{5 characters}\\\text{(space, 3 digits,}\\\text{space)}}}\ \underbrace{\text{CHR\$(CODE)}}_{\substack{\text{1 graphics}\\\text{character}}}\ \underbrace{\text{STRING\$(2, 32)}}_{\text{2 spaces}}$$

5 + 1 + 2 = 8 characters

The TRS-80's screen width is 64 characters. So, after printing eight codes and graphics characters, the TRS-80 reaches the right end of the screen and the cursor moves to position zero (0) at the next line.

Aha! Now we see what line 230 does. It prints an empty line after every eighth execution of line 220. Try the program *without* line 230 just to see what happens.

BY THE DRAGON

Notes to Our Advertisers

- Look for our glossy cover beginning next month (Mar-Apr)
- Themes of upcoming issues: Mar-Apr **GAMES AND RECREATIONS**; May-Jun **6502 PROCESSORS**; Sep-Oct **LEARNING**; Nov-Dec **MUSIC AND ART**
- **RECREATIONAL COMPUTING MARKETPLACE**—small ads page coming in Mar-Apr issue

Write or call our Advertising Manager for further information, deadlines, samply copy. People's Computer Company, 1263 El Camino Real, Menlo Park, CA 94025 (415) 323-3111.

Announcements

Hardware

Erase-Sure. Instantly erases previous programs or recordings from cassettes, and Rapid Rewinder rewinds cassettes in just seconds. Produce quality recordings, save wear on equipment. Each is $24.50 plus $1.50 shipping. Magnesonics Sales, P.O. Box 758, Ventura, CA 93001, (805) 642-3092.

Stringy Floppy Mass Storage System. High performance system for micros. Suitable for numerous applications. Package includes drive unit, tape heads, operating program, etc. Base system package is $249.50. Exatron, 3555 Ryder St., Santa Clara, CA 95051, (408) 737-7111.

Double Density Floppy Disk Interface. A component that enhances existing disk storage capacities with only minimum reconfiguration of existing systems. BASIC input/output system software for CP/M on single-density diskette. With BIOS for CP/M on diskette, priced at $425. Tarbell Electronics, 950 Dovlen Place, Suite B, Carson, CA 90746, (213) 538-4251.

PERCOM Plug-In Adapter. This adapter is for TRS-80 and SWTP MP-F mini-floppy disk controllers for data separation function and reducing data read errors. Called the separator, easy to install and use. $29.95. Percom Data Co., 211 N. Kirby, Garland, Texas 75042, (214) 272-3421.

SUPER DAZZLER. Super Dazzler Interface is a high-resolution graphics interface for Cromemco computer systems. Displays color or black-and-white images with up to 756 by 484 point resolution. Super Dazzler is $595, 16K two-port memory card $795. Cromemco, Inc., 280 Bernardo Avenue, Mountain View, CA 94043, (415) 964-7400.

INTROL/X-10 for Apple Computers. Introl/X-10 allows users to remotely control 110 volt A.C. electrical devices by commands sent over existing building wiring. Up to 16 remote modules may be controlled. Base unit $279. Mountain Hardware, Inc., 300 Harvey West Blvd., Santa Cruz, CA 95060.

Skyles PAL. This is a high-performance serial printer featuring 40 character-per-second printing with a full 96 character set. Full line buffering and bidirectional look-ahead printing, 5 x 7 dot-matrix. $450. Skyles Electric Works, 10301 Stonydale Drive, Cupertino, CA 95014 (408) 735-7891.

Software

Hardcopy Graphics Program. This program for the PET computer has full graphics capability for Houston Instrument's HIPLOT plotter. Programs in BASIC are available for PET, TRS-80 and APPLE II, drive the plotter through a RS-232 interface. Plotter retails for $1085, the programs are $50 and $75. Memory requirement is 16K bytes. WEST COAST CONSULTANTS, 1775 Lincoln Blvd, Tracy, CA. 95376, (209) 835-1780.

Dungeon Explorer. This game, by Matthew D. Kiriazis, is an adventure game for the TRS-80. It is a single player game full of fantastic combat and adventure, based upon Dungeons and Dragons. Available for $8.50. Software Exchange, 2681 Peterboro, W. Bloomfield, MI 48033.

Dunjonquest. A fantasy adventure series for the TRS-80 and PET microcomputers, includes Temple of Apshai, Starfleet Orion and Invasion Orion. Cassette or disk, $19.95 and $24.95 from Automated Simulations, P.O. Box 4232, Mountain View, CA 94040.

Classic Games Series. A series of machine language software for the TRS-80. Features BACK-40, our Backgammon program, Z-Chess and Dr. Chips with graphic display of unrivaled quality, clarity and ease of play. $14.96 and $17.95. The Software Association, P.O. Box 58365, Dept. RC, Houston, TX 77058 (713) 482-0883.

Software Library. A set of fifty programs consisting of games, teaching programs and practical applications. Included are Trek 3, Backgammon, Checkbook, Condrum, Arena and Calendar. Introductory special $49.95 on cassette or $59.95 on diskette. Dr. Daley's Software, 425 Grove Ave., Berrien Sprints MI 49103, (616) 471-5514.

MICROSKETCH II. The first of a series of top quality programs for the TRS-80, Level II or Disk BASIC. The Main System includes 49 commands and can create an infinite variety of graphics. Includes five subsystems. Only $3.95. International Data Services, 340 West 55th St., New York, N.Y. 10019, (212) 757-8046.

Timeshared Operating System. For the 8080 and Z80 series of micros, called Chaos II. Features multiprogramming capability, several programming languages, complete file/directory system, etc. Object code disk $300, Source listings $500. Computer Systems Design Group, 3632 Governor Dr., San Diego, CA 92122.

Scientific Report. Computer assisted research report program on avians (birds) for Apple II. Enjoyable and colorful program that teaches about birds. Purchase supports research. $10 for two programs. American Avicultural, 3268 Watson Road, Saint Louis, MO 63139, (314) 645-4431.

Dakin5 Programming Aids. A package of seven programming aids mainly for Apple II. Includes Lister, Peeker, Cruncher, Diskette Copy, Prompter, Calculator and Text File Copy. Price is $39.95. Visit dealer or contact Dakin5 Corporation, 7475 Dakin Street, Suite 507, Denver, CO 80221, (303) 426-6090.

Business Software Packages. For the TRS-80 Model II computer. Extensively modified programs to use expanded hardware capabilities. Send $10 for each manual, and two SASE's for catalongs. Micro Architect, 96 Dothan St., Arlington, MA 02174.

Disc Drive Timing Program. For TRS-80 and Apple II. Keep track of disc drive motor speed on a routine basis. Works on any disc drive. Cassette or diskette, $14.95 and $19.95. DISCO-TECH, P.O. Box 11129, Santa Rosa, CA 95406.

Mathematics in BASIC Series. The program tapes solve various mathematical operations. They are practical working aids for professionals and applied learning tools for educators. Hayden Book Company, Inc., 50 Essex Street, Rochelle Park, New Jersey 07662, (201) 843-0550.

Super-Compiler. For the 6502, called XPLO. A structured language. Run programs 2.5 to 16 times faster. Versions for 20K Apple II, KIM, TIM and SYM systems are available for under $70. Send for free catalog. The 6502 Program Exchange, 2920 Moana, Reno, NV 89509.

Software for PET. Programs provide the PET with nineteen functions, including: standard and custom renumbering, compact, trace, append and merge, and additional disk and tape related functions. Tape or disk $100, or on ROM for $200. National Artificial Intelligence Laboratory, P.O. Box F, Mobile, AL 36601, (205) 433-5529.

Educational Software

Educational Software Series. For the PET, Apple II, and TRS-80. Three new titles:
Reading Comprehension: What's Different? presents logical problems; *Minicrossword* builds vocabulary and spelling skills; *Word Skills 1—Prefixes* presents common prefixes and the words they appear in.
From computer stores or Program Design, 11 Idar Court, Greenwich, CT 06830, (203) 661-8799.

MICRO-REDY project. A federally funded effort to develop computer assisted instruction for the educationally disadvantaged elementary school student, using Apples. Send any information or ideas you have to share; you may have CAI programs resulting from the project. Barry Cole, c/o Sacramento City Unified School District, Box 2271, Sacramento, CA 95810.

Learning Programs. 21 programs for Apple II for elementary and junior high level, including computer literacy, language arts, math, time-telling and games. Visual, verbal and auditory reinforcements. $15.75 to $19.50. Edutak Corp., P.O. Box 11354, Palo Alto, CA 94306.

Softswap. A project of the Oregon Council of Computer Educators, seeks to establish groups of users of same equipment who wish to share educational programs. Contact: Fred Beisse, the Real Oregon Computer Company, 207 West 10th Ave., Eugene, OR 97401.

MicroGnome's CAIWARE. A Software system for authoring and using Computer Assisted Instruction on the 16K TRS-80 with Level II BASIC. CAIWARE is an aid, not a replacement for the teacher or the textbook. Program is $24.95. Fireside Computing, Inc., 5843 Montgomery Road, Elkridge, MD 21227, (301) 796-4165.

Educational Program Information Center (EPIC) has been established by the Apple Education Foundation to encourage the production and use of microcomputer-based learning materials. For more information: EPIC Review, Apple Education Foundation, 20605 Lazaneo Drive, Cupertino, CA 95014.

Publications

TRS-80 Software Source Directory. ComputerMat lists over 5,000 TRS-80 software programs that are available from over 380 vendors. Designed to help locate software for home, educational, personal and professional areas. $6. per issue, foreign orders add $2. ComputerMat, Box 1664, Lake Havasu, AZ 86403, (602) 855-3357.

New Software Publication. *80 Software Critique* is a collection of reviews of TRS-80 cassette software. The reviews are detailed and represent many hours of actual use. Programs reviewed include games, simulations, educational programs, music programs and others. One year is $24, single copy $7. *80 Software Critique*, P.O. Box 134, Waukegan, IL 60085.

Computer Coin Games. By Joe Weisbecker, a new instructional book which employs games, illustrations and clever copy to teach readers of all ages about computers. A supplement to other teaching materials. $3.95 softbound. 96 pages. Published by Creative Computing Press, P.O. Box 789-M, Morristown, N.J. 07960.

Computing Teacher. A high-quality, bi-monthly journal for computer educators who wish to keep up with new developments in the field in association with national group called Computer Using Educators. Don McKall, Independence MS, 1776 Education Park Dr., San Jose, CA 95133.

Catalog of Microcomputer Programming. A 32-page compendium listing over 300 programs for PET, Apple, and TRS-80 microcomputers, covering uses from science, education, and business to entertainment and use in the home. Inquire at computer stores or write Instant Software Inc., Catalog Dept., Peterborough, NH 03458.

Other

Greater Baltimore Hamboree and Computerfest. An electronic experimenters show and swapfest, will be held on Sunday, March 30, 1980 at the Maryland State Fairgrounds at Timonium. Exciting computer displays and exhibits, dealers and hobbyists, door prizes, etc. For information or reservations: Joseph A. Lochte Jr., 2136 Pine Valley Dr., Tomonium, MD 21093.

DunDraCon. The science-fiction, fantasy, and role-playing game convention is back again for the fifth year, bigger and better than ever! It will be at the Villa Hotel in San Mateo, 4000 El Camino on the 16th, 17th and 18th of February 1980. DunDraCon V, 386 Alcatraz, Oakland, CA 94618.

Monterey Bay Users Group. (MBUG) for the TRS-80. We meet on the last Friday of every month at the Naval Postgraduate School in Monterey. Our group has as varied interests as there are uses for the TRS-80. The Monterey Bay Users Group, P.O. Box 1242, Seaside, CA 93955.

Crescent City Computer Club. Holds Friday meetings at 8 pm in Room 2120, UNO Science Bldg., P.O. Box 1097, University of New Orleans, LA 70122.

Research and Development Project. At University of Oregon, to provide a non-threatening first experience with micros for K-12 teachers. In the self-instructional, laboratory-type course, the learner will use a PET and explore more than 50 programs. Project director is Dan Isaacson, Computer Center, Univ. of Oregon, Eugene, OR 97403.

Special!

A FILM! One Pass, Inc. is pleased to announce that the film "Don't Bother Me, I'm Learning" will be shown over the Public Broadcasting Network on Sunday, January 6, 1980 at 6:00 p.m. This timely production will present a fascinating view of the microelectronic revolution and its impact on the learning process of children.

This film was taped at the Lawrence Hall of Science in Berkeley, at one public elementary school and two private schools in the San Francisco Bay Area, at IBM Research, and at a futuristic home in Hillsborough, California. The participants in the show all advocate computer education, but they come to the subject from a variety of perspectives. Art Luehrmann of Lawrence Hall; and Dean Brown, an educator and consultant; Ramon Zamora of People's Computer Company; Mary Laycock and Pat Tubbs, educators; and parents and kids.

The technical consultant for this production is Bob Albrecht (also of PCC), an author and lecturer, who was described in *Time* (February 20, 1978) as being "a pioneer in electronic education." "Don't Bother Me, I'm Learning" was produced by David Shepardson and directed by Karen Carlson. The executive producer is Steve Michelson. Major funding for this PBS broadcast has been provided by the Bell & Howell Corporation.